Defense Information Systems Agency
Center for Standards

DEPARTMENT OF DEFENSE TECHNICAL ARCHITECTURE FRAMEWORK FOR INFORMATION MANAGEMENT

Volume 3: Architecture Concepts and Design Guidance

Version 3.0

30 April 1996

DRAFT SF 298

1. Report Date (dd-mm-yy)
30 April 1996

2. Report Type

3. Dates covered (from... to)

4. Title & subtitle
Department of Defense Technical Architecture Framework for Information Management. Volume 3: Architecture Concepts and Design Guidance.

5a. Contract or Grant #

5b. Program Element #

6. Author(s)

5c. Project #

5d. Task #

5e. Work Unit #

7. Performing Organization Name & Address

8. Performing Organization Report #

9. Sponsoring/Monitoring Agency Name & Address
Defense Information Systems Agency
Center for Standards
10701 Parkridge Blvd
Reston, VA 20191

10. Monitor Acronym

11. Monitor Report #

12. Distribution/Availability Statement Approved for Public Release: Distribution is Unlimited

13. Supplementary Notes

14. Abstract

15. Subject Terms

[redacted]

16. Report
Unclass

17. Abstract
Unclass

18. This Page
Unclass

19. Limitation of Abstract

20. # of Pages

21. Responsible Person (Name and Telephone #)

Marilyn McLaughlin
(703) 735-3563

FOREWORD:
ABOUT THIS DOCUMENT

This edition of the Technical Architecture Framework for Information Management (TAFIM) replaces Version 2.0, dated 30 June 1994. Version 3.0 comprises eight volumes, as listed on the following configuration management page.

TAFIM HARMONIZATION AND ALIGNMENT

This TAFIM version is the result of a review and comment coordination period that began with the release of the 30 September 1995 Version 3.0 Draft. During this coordination period, a number of extremely significant activities were initiated by DoD. As a result, the version of the TAFIM that was valid at the beginning of the coordination period is now "out of step" with the direction and preliminary outcomes of these DoD activities. Work on a complete TAFIM update is underway to reflect the policy, guidance, and recommendations coming from theses activities as they near completion. Each TAFIM volume will be released as it is updated. Specifically, the next TAFIM release will fully reflect decisions stemming from the following:

- The DoD 5000 Series of acquisition policy and procedure documents
- The Joint Technical Architecture (JTA), currently a preliminary draft document under review.
- The C4ISR Integrated Task Force (ITF) recommendations on Operational, Systems, and Technical architectures.

SUMMARY OF MAJOR CHANGES AND EXPECTED UPDATES

This document, Volume 3 of the TAFIM, contains minor substantive changes from Volume 3 of Version 2.0, most of which are intended to resolve internal inconsistencies or to bring the guidance provided in this volume more in line with current policies. Work remains to be done to fully reflect the impact of the documents and decisions noted above; this edition of the TAFIM has been released to serve as a baseline and to make available throughout the DoD community the additions and modifications that have been implemented to date.

A NOTE ON VERSION NUMBERING

A version numbering scheme approved by the Architecture Methodology Working Group will control the version numbers applied to all future editions of TAFIM volumes. Version numbers will be applied and incremented as follows:

- This edition of the TAFIM is the official Version 3.0.
- From this point forward, single volumes will be updated and republished as needed. The second digit in the version number will be incremented each time (e.g., Volume 7 Version

3.1). The new version number will be applied only to the volume(s) that are updated at that time. There is no limit to the number of times the second digit can be changed to account for new editions of particular volumes.

- On an infrequent basis (e.g., every two years or more), the entire TAFIM set will be republished at once. Only when all volumes are released simultaneously will the first digit in the version number be changed. The next complete version will be designated Version 4.0.
- TAFIM volumes bearing a two-digit version number (e.g., Version 3.0, 3.1, etc.) without the DRAFT designation are final, official versions of the TAFIM. Only the TAFIM program manager can change the two-digit version number on a volume.
- A third digit can be added to the version number as needed to control working drafts, proposed volumes, internal review drafts, and other unofficial releases. The sponsoring organization can append and change this digit as desired.

Certain TAFIM volumes developed for purposes outside the TAFIM may appear under a different title and with a different version number from those specified in the configuration management page. These editions are not official releases of TAFIM volumes.

DISTRIBUTION

Version 3.0 is available for download from the DISA Information Technology Standards Information (ITSI) bulletin board system (BBS). Users are welcome to add the TAFIM files to individual organizations' BBSs or file servers to facilitate wider availability.

This final release of Version 3.0 will be made available on the World Wide Web (WWW) shortly after hard-copy publication. The Defense Information Systems Agency (DISA) is also investigating other electronic distribution approaches to facilitate access to the TAFIM and to enhance its usability.

TAFIM Document Configuration Management Page

The **latest authorized versions** of the TAFIM volumes are as follows:

Volume 1:	Overview	3.0	30 April 1996
Volume 2:	Technical Reference Model	3.0	30 April 1996
Volume 3:	Architecture Concepts & Design Guidance	3.0	30 April 1996
Volume 4:	DoD SBA Planning Guide	3.0	30 April 1996
Volume 5:	Program Manager's Guide for Open Systems	3.0	30 April 1996
Volume 6:	DoD Goal Security Architecture	3.0	30 April 1996
Volume 7:	Adopted Information Technology Standards	3.0	30 April 1996
Volume 8:	HCI Style Guide	3.0	30 April 1996

Working drafts may have been released by volume sponsors for internal coordination purposes. It is not necessary for the general reader to obtain and incorporate these unofficial, working drafts.

Note: Only those versions listed above as authorized versions represent official editions of the TAFIM.

This page intentionally left blank.

CONTENTS

FIGURES

1.0 INTRODUCTION

The Technical Architecture Framework for Information Management (TAFIM) characterizes an information system as composed of data, mission-specific applications, and a technical infrastructure consisting of support applications, application platforms, and communications networks. This document presents technical architecture concepts and design guidance for information systems in the Department of Defense (DoD). As part of the TAFIM, this volume provides guidance for the evolution of the DoD's technical infrastructure in support of specific mission requirements. The data and mission-specific software architectures are critical elements in information system development. Guidance on their development and use will be provided in separate documents outside of the TAFIM.

1.1 SCOPE

Volume 3 provides concepts and design guidance that will help architects, integrators, and system designers to develop information systems technical architectures. These concepts and guidance should be considered in the context of the Technical Reference Model presented in Volume 2.

The contents of this volume contrast with the TAFIM Volume 2, which describes services and interfaces between entities. This volume addresses components and the allocation of services to the components.

1.2 DOCUMENT ORGANIZATION

Volume 3 of the TAFIM consists of three sections and four appendices. Section 2 discusses several architecture concepts of interest to architects, designers, and developers. Section 3 presents design guidance based on availability and maturity of technology. Appendix A provides acronyms and definitions. Appendix B provides a definition and discussion of open systems. Appendix C contains references. Appendix D contains a template for submitting comments on this volume.

This page intentionally left blank.

2.0 ARCHITECTURE CONCEPTS

The DoD vision described in Volume 1 depicts a future information technology environment that includes:

- Shared global databases
- Shared utility services
- Centrally managed and operated backbone network
- Distributed data
- Distributed processes
- Standard user interface
- Transparent information, computing, and information utility
- Individual tailoring of information resources.

The new DoD architectural framework supports an orderly migration from existing legacy systems to DoD standard systems operating in a distributed computing environment that incorporates the above features. Over time, systems will be reengineered or developed to conform to the architecture concepts and standards in the TAFIM. As this occurs, a distributed computing environment will evolve, where processing nodes are constructed to provide services to meet the requirements of the DoD community.

Figure 2-1 depicts a distributed computing environment, which includes platforms and, optionally, support applications and mission-area applications. The external environment shown in the figure consists of entities that are external to the application software and the platform (e.g., users, communications networks). The actual features of an implementation will be dictated by functional requirements and processing efficiency. The enterprise backbone network in this distributed computing environment will provide end-to-end communications services that connect all of the processing nodes, down to an individual's workstation. Through the network, authorized users and applications will have access to all required data and processing resources without having to know the location of the resources. This will include access to shared enterprise global databases and utility services by distributed applications and fixed and mobile users. Resources will be provided through a transparent combination of local and remote processes. Distribution of redundant enterprise data and processing resources across multiple processing nodes will provide processing efficiency, reliability, and survivability.

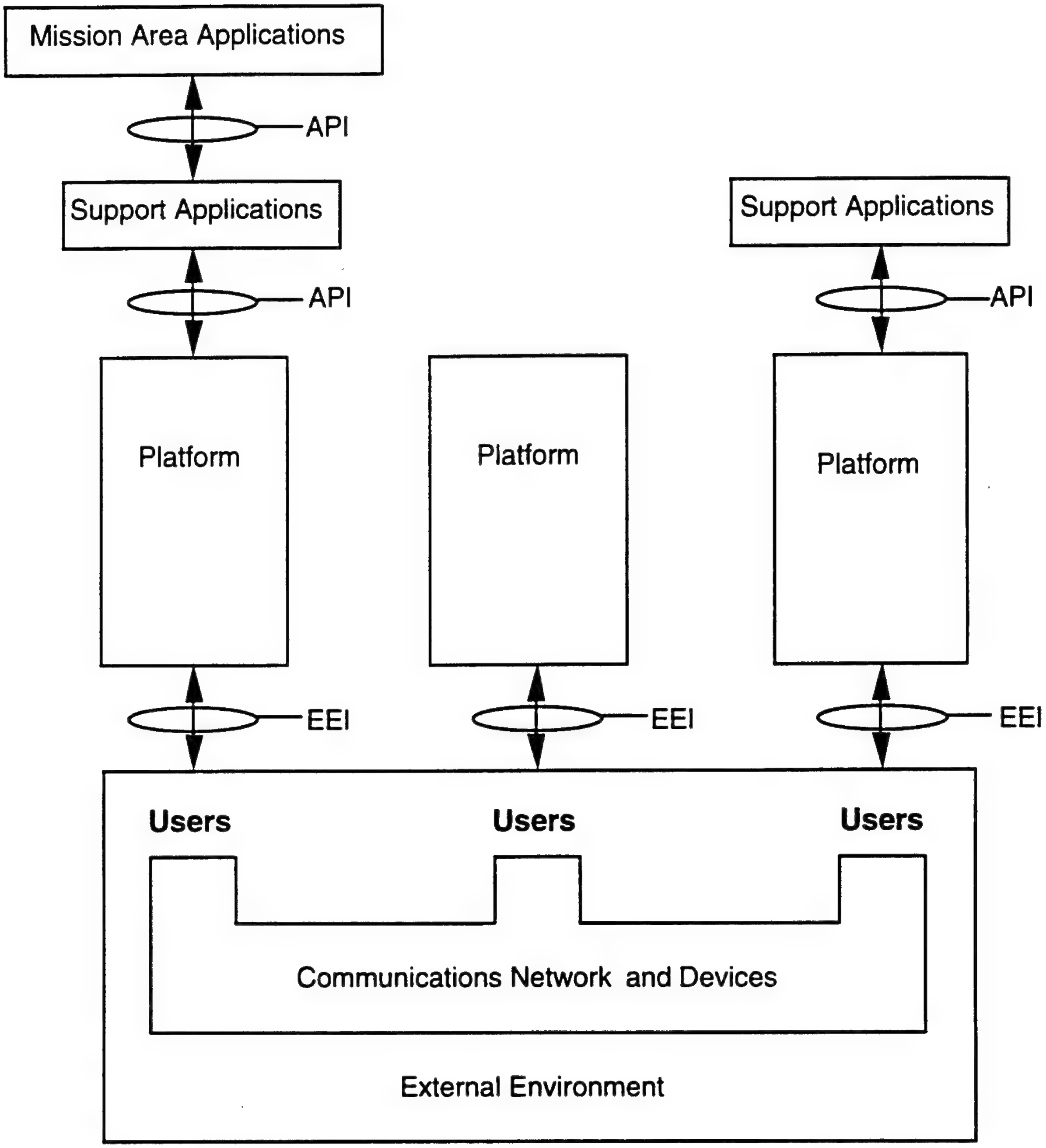


Figure 2-1. Distributed Computing Context

2.1 ARCHITECTURE DEFINITIONS

This section sets the stage for the architecture concepts to be described. It provides basic definitions within the context of a model for architectures.

2.1.1 Architecture Model

A technical architecture defines components, interfaces, services, and the framework within which they interoperate. Components provide either information processing or communications services. A component provides a complete service or part of a service. A component may also provide more than one service. Interfaces link components so that they may interoperate. Figure 2-2 depicts a model of these relationships.

Figure 2-3 depicts service components and their interfaces. The TAFIM provides guidance on the following interfaces: a) between applications (mission-area and support applications) and service components, b) between separate service components, and c) between service components and the external environment.

Services are invoked through an interface, which defines the access rules. Two types of interfaces are described in the Technical Reference Model: an application program interface (API), which defines the rules and protocols used by an application to invoke a service; and an external environment interface (EEI), which defines the rules and protocols for invoking the external environment services. EEI services are provided to support users, peripherals, and remote processors. Volume 2 defines an API as the interface that enables applications to invoke application platform services. To satisfy DoD Information Management (IM) requirements, the TAFIM has applied the definition of an API to any service provided to an application through a programming interface. This interpretation was necessary to meet two distinct requirements.

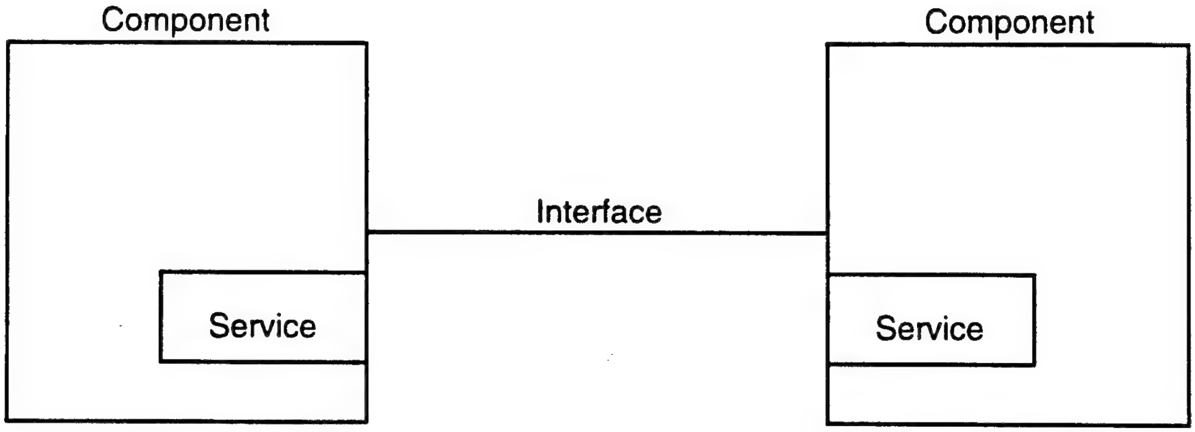


Figure 2-2. Model of Information System Architecture

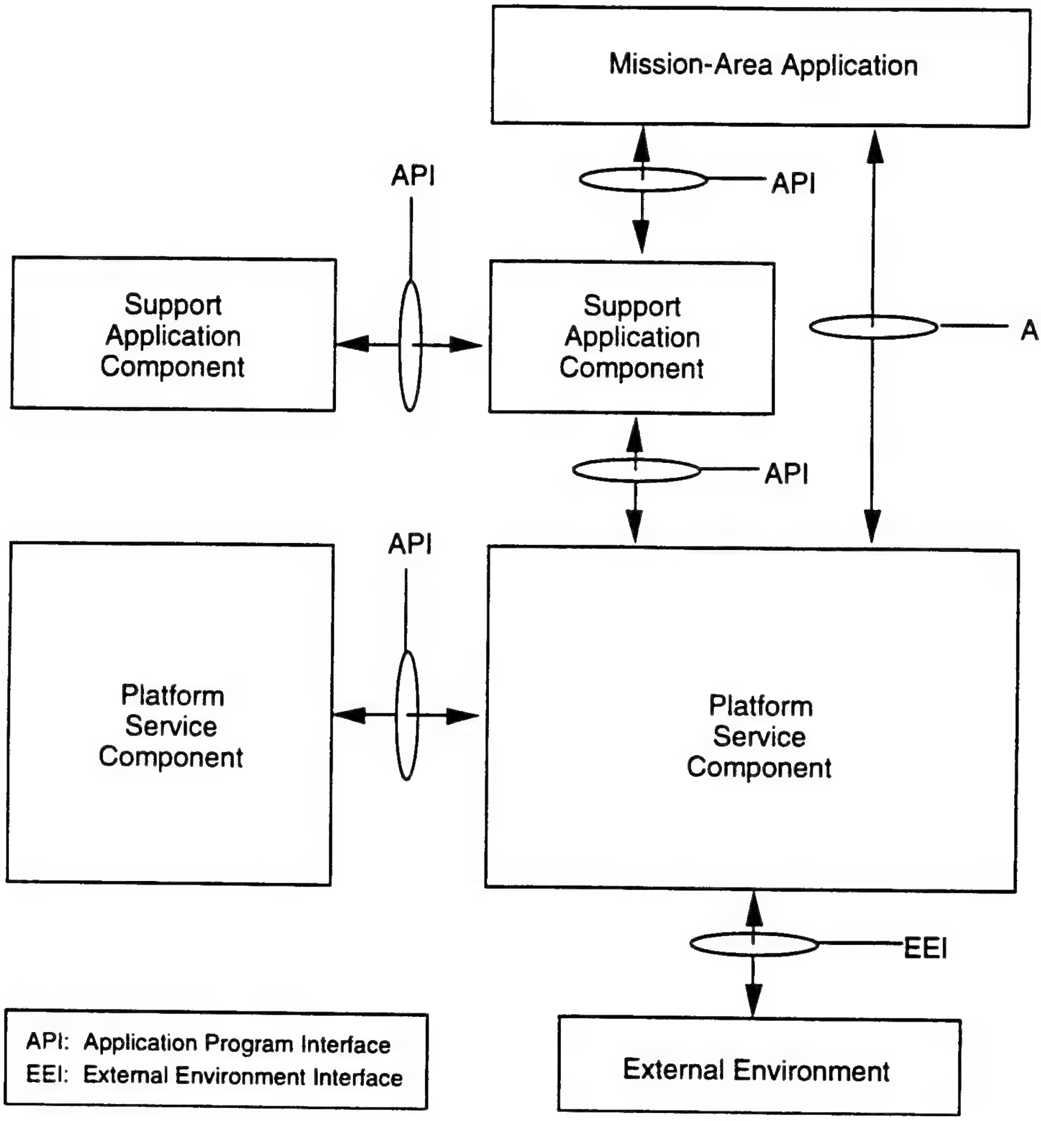

Mission-Area Application
API
API
Support Application Component
Support Application Component
API
API
API
Platform Service Component
Platform Service Component
EEI
API: Application Program Interface
EEI: External Environment Interface
External Environment

Figure 2-3. Technical Architecture Service Context

First, it supports the use of services not provided by the application platform. Recognizing that many reusable services are not covered under the platform service categories, the TAFIM has split the applications into mission-area applications and support applications. The support applications provide common reusable services, such as word processing and electronic mail (E-mail), to mission-area applications and other support applications. To support mission-area and support application portability, DoD has a requirement for standard application interfaces to these services. Applying the definition of the API to address this interface supports the potential future migration of services between the support applications and platforms.

Second, this expansion supports the distribution of computing and communications resources throughout the network. The use of one platform component's service by another platform component is defined as a system internal interface (SII) by POSIX P1003.0. The DoD requirement is for platform components to use the same API that an application uses when requesting services from other platform components. For example, if an Information Resource Dictionary System (IRDS)-compliant dictionary has a requirement to store data in a database, it should use the Structured Query Language (SQL) API to invoke the services of a Relational Database Management System (RDBMS). This will minimize unique dependencies between platform components, enhancing the capability to replace one platform component with another. It will also provide DoD with the maximum flexibility possible in distributing computing and communication resources throughout the network.

2.1.2 Architecture Views

Depending on the area of responsibility of the architect or designer, an architecture may be viewed from different perspectives. For example, the designer responsible for computing perceives the architecture with a different focus than the designer responsible for data management. The architect responsible for the overall system has yet another focus. The views presented in the remaining subsections of Section 2 (2.2, 2.3, 2.4, 2.5) describe architecture concepts from different perspectives. Each of these views addresses components, interfaces, and allocation of services critical to the view.

2.2 COMPUTING VIEW

This view of the technical architecture focuses on computing models that are appropriate for a distributed computing environment. To support the migration of legacy systems, the section also presents models that are appropriate for a centralized environment. The definitions of many of the computing models (e.g., host-based, master-slave, and three-tiered) historically preceded the definition of the client/server model, which attempts to be a general-purpose model. In most cases the models have not been redefined in the computing literature in terms of contrasts with the client/server model. Therefore, some of the distinctions of features are not always clean. In general, however, the models are distinguished by the allocation of functions for an information system application to various components (e.g., terminals, computer platforms). These functions that make up an information system application are presentation, application function, and data management.

2.2.1 Client/Server Model

Client/server processing is a special type of distributed computing termed cooperative processing because the clients and servers cooperate in the processing of a total application (presentation, functional processing, data management). In the model, clients are processes that request services, and servers are processes that provide services. Clients and servers can be located on the same processor, different multiprocessor nodes, or on separate processors at remote locations. The client typically initiates communications with the server. The server typically does not initiate a request with a client. A server may support many clients and may act as a client to another server. Figure 2-4 depicts the basic client/server model.

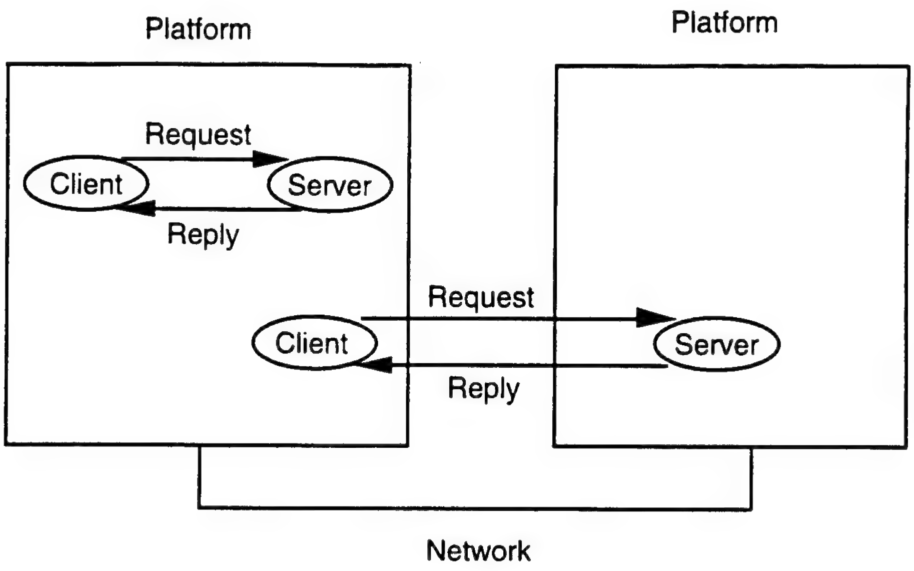


Figure 2-4. Basic Client/Server Model

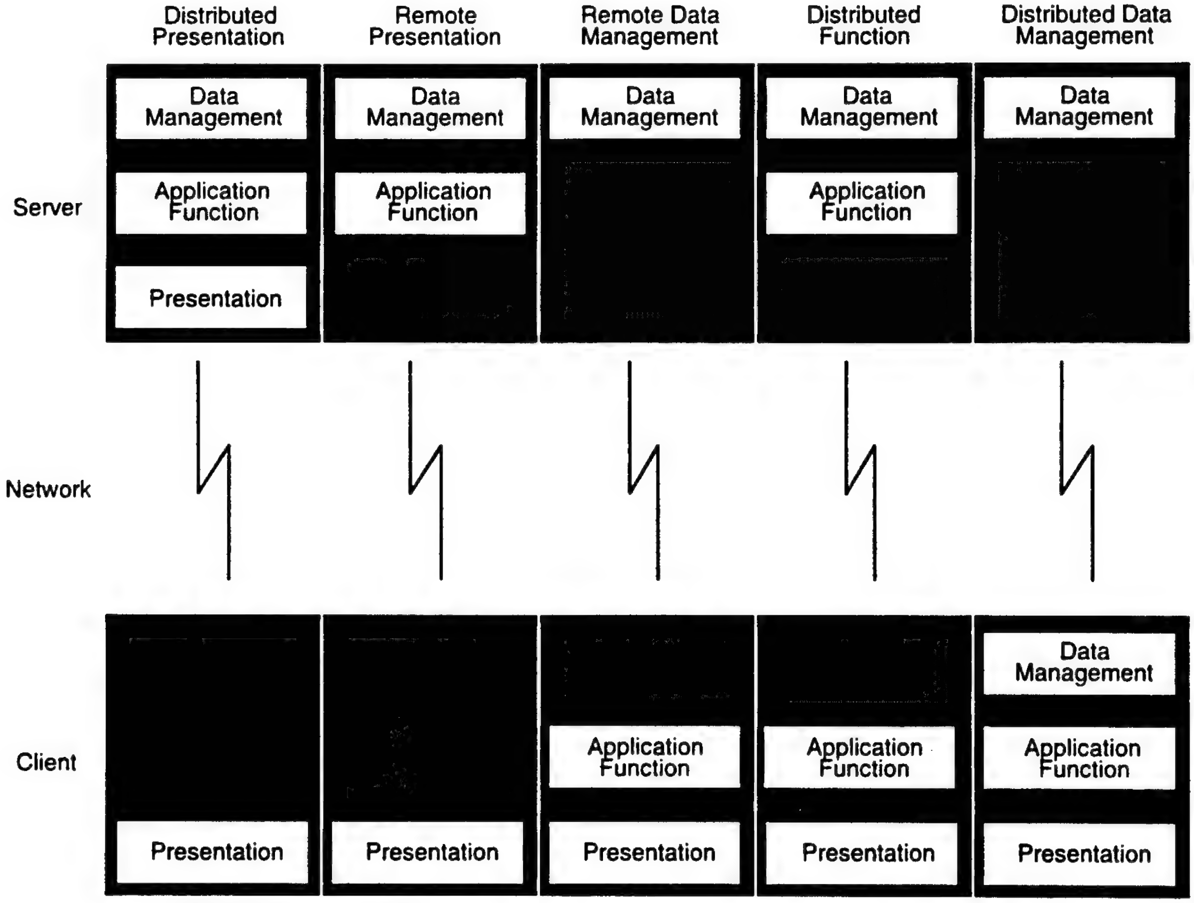
Clients tend to be generalized and can run on one of many nodes. Servers tend to be specialized and run on a few nodes. Clients are typically implemented as a call to a routine. Servers are typically implemented as a continuous process waiting for service requests (from clients). Many client/server implementations involve remote communications across a network. However, nothing in the client/server model dictates remote communications, and the physical location of clients is usually transparent to the server. The communication between a client and a server may involve a local communication between two independent processes on the same machine.

An application program can be considered to consist of three parts–the application function, the presentation, and the data management. In general, any of these can be assigned to either a client or a server. The assignment of each of these program parts to clients and servers can define client/server configurations. The following are five client/server configurations, which demonstrate the flexibility of the client/server model in implementing distributed paradigms. The terms "remote" and "distributed" are from the perspective of the application function portion of the processing:

- Distributed Presentation
- Remote Presentation
- Remote Data Management
- Distributed Function
- Distributed Data Management.

These five client/server configurations, which are based on a Gartner Group Report, are frequently cited in computing literature, but some sources find that they do not represent the current state of commercial-off-the-shelf (COTS) offerings. For example, the Defense Information Systems Agency's (DISA) *Client Server Migration Guidance* presents three models as alternatives to the above popular configurations. They are Presentation Logic Functions, Business Logic Functions, and Data Management Functions. Their rationale is as follows. The distinctions between remote and distributed presentation, data management, and distributed function do not relate easily to COTS products. As an example, the remote data management model states that the presentation functions reside entirely in a client. In practice, virtually every implementation places some functions, however small, in a server. This places these implementations into the distributed presentation model category. Similar situations occur for the other models.

Another client/server configuration growing in popularity is the multitiered architecture. The multitiered architecture and the five popular client/server configurations that are listed above are discussed in the following sections. The five client/server configurations are presented in Figure 2-5.



Source: Gartner Group

Figure 2-5. Client/Server Configurations

2.2.1.1 Distributed Presentation

This model distributes responsibility for presentation between the client and the server. An architecture that implements this model is the X Window architecture. In such an implementation, X terminals are used as client platforms, which contain presentation functions. All other functions (application function and data management), including additional presentation capability, are on the server.

2.2.1.2 Remote Presentation

This configuration separates presentation logic from functional logic by locating the entire presentation function on the client workstation. The client is responsible for the user interface, accepting and validating user input, and sending requests to and receiving results of requests from the server. The advantage of this model is that client and servers are separated by a network, keeping presentation logic off the network. In this scenario, clients request data management and application functional processing services from servers.

2.2.1.3 Remote Data Management

In this configuration, a central server specializes in data management, which might include data security, integrity, and processing database requests from the client. The management of data is separate from the application. This model can be used to support central subject area databases serving one or more remote clients. An example configuration is a database machine or database server attached to clients on workstations.

2.2.1.4 Distributed Function

In this configuration, multiple servers provide specific application processing functions for client applications. The advantages offered through the distribution of application functions include the reduction of redundant code, centralized management and operation of complex processing functions, the ability to distribute some application functions closer to the end user, and the ability to configure and tune specialized servers for maximum processing efficiency. Examples of servers that provide distributed application functions include mail servers, print servers, transaction processors, communication servers, mission-area application servers, and directory servers.

2.2.1.5 Distributed Data Management

In this configuration, the responsibility for data management is split among more than one server. When one data management server, which may or may not be local to the client application, cannot satisfy a request for data, it in turn becomes a client to another data management server that is capable of satisfying the original request. The original client application is unaware that more than one server participated in processing the request. This variation can be introduced during application consolidation and migration, where data is distributed across multiple legacy databases. It can also be used to support an environment where a logical subject area database is spread over several physical databases. An example configuration is a distributed database on more than one platform.

2.2.1.6 The Multitiered Architecture

All of the client/server configurations presented so far in this section (2.2.1) show functions (presentation, application logic, and data management) distributed over two virtual platforms. These can be considered two-tiered architectures. Multitiered client/server architectures with three or more tiers have been proposed and are gaining in popularity.

In multitiered architectures, functions are distributed over multiple virtual (or logical) platforms. These architectures accommodate the partitioning of applications so that user interfaces reside on the user's platform, functional services reside on one or more other networked platforms, and data and legacy systems reside on additional networked platforms.

2.2.2 Host-Based Model

The host-based model is an approach that provides centralized processing on a host machine – that is, it provides no distributed processing. The typical configuration is a mainframe with attached dumb terminals. The central computer does all of the processing (e.g., presentation, application functional processing, data management). Figure 2-6 presents an example host-based configuration.

2.2.3 Master-Slave and Hierarchic Models

In this model, slave computers are attached to a master computer. In terms of distribution, the master-slave model is one step up from the host-based model. Distribution is provided in one direction–from the master to the slaves. The slave computers perform application processing only when directed to by the master computer. In addition, slave processors can perform limited local processing, such as editing, function key processing, and field validation. A typical configuration might be a mainframe as the master with personal computers (PC) as the slaves acting as intelligent terminals, as illustrated in Figure 2-6.

The hierarchic model is an extension of the master-slave model with more distribution capabilities. In this approach, the top layer is usually a powerful mainframe, which acts as a server to the second tier. The second layer consists of local area network (LAN) servers and clients to the first layer as well as servers to the third layer. The third layer consists of PCs and workstations. This model has been described as adding true distributed processing to the master-slave model. Figure 2-6 shows an example hierarchic model in the third configuration.

2.2.4 Peer-to-Peer Model

In the peer-to-peer model there are coordinating processes. All of the computers are servers in that they can receive requests for services and respond to them; and all of the computers are clients in that they can send requests for services to other computers. In current implementations, there often are redundant functions on the participating platforms.

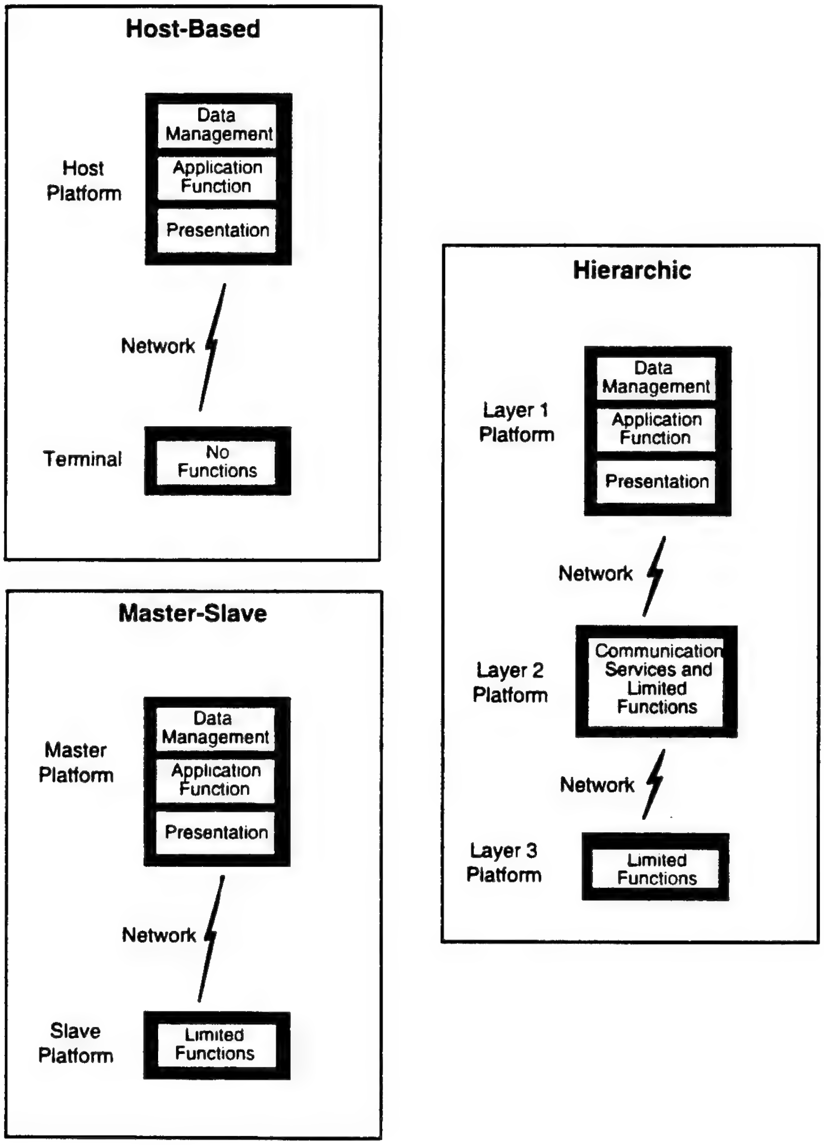


Figure 2-6. Host-Based, Master-Slave, and Hierarchic Models

Attempts have been made to implement the model for distributed heterogeneous (or federated) database systems. This model could be considered a special case of the client/server model, in which all platforms are both servers and clients. Figure 2-7 (A) shows an example peer-to-peer configuration in which all platforms have complete functions.

2.2.5 Distributed Object Management Model

In this model the remote procedure calls typically used for communication in the client/server and other distributed processing models are replaced by messages sent to objects. The services provided by systems on a network are treated as objects. A requester need not know the details of how the object is configured. The approach requires: 1) a mechanism to dispatch messages; 2) a mechanism to coordinate delivery of messages; and 3) applications and services that support a messaging interface. This approach does not contrast with client/server or peer-to-peer models but specifies a consistent interface for communicating between cooperating platforms. It is considered by some as an implementation approach for client/server and peer-to-peer models. Figure 2-7 presents two distributed object model examples. Example B shows how a client/server configuration would be altered to accommodate the distributed object management model. Example C shows how a peer-to-peer model would be altered to accomplish distributed object management.

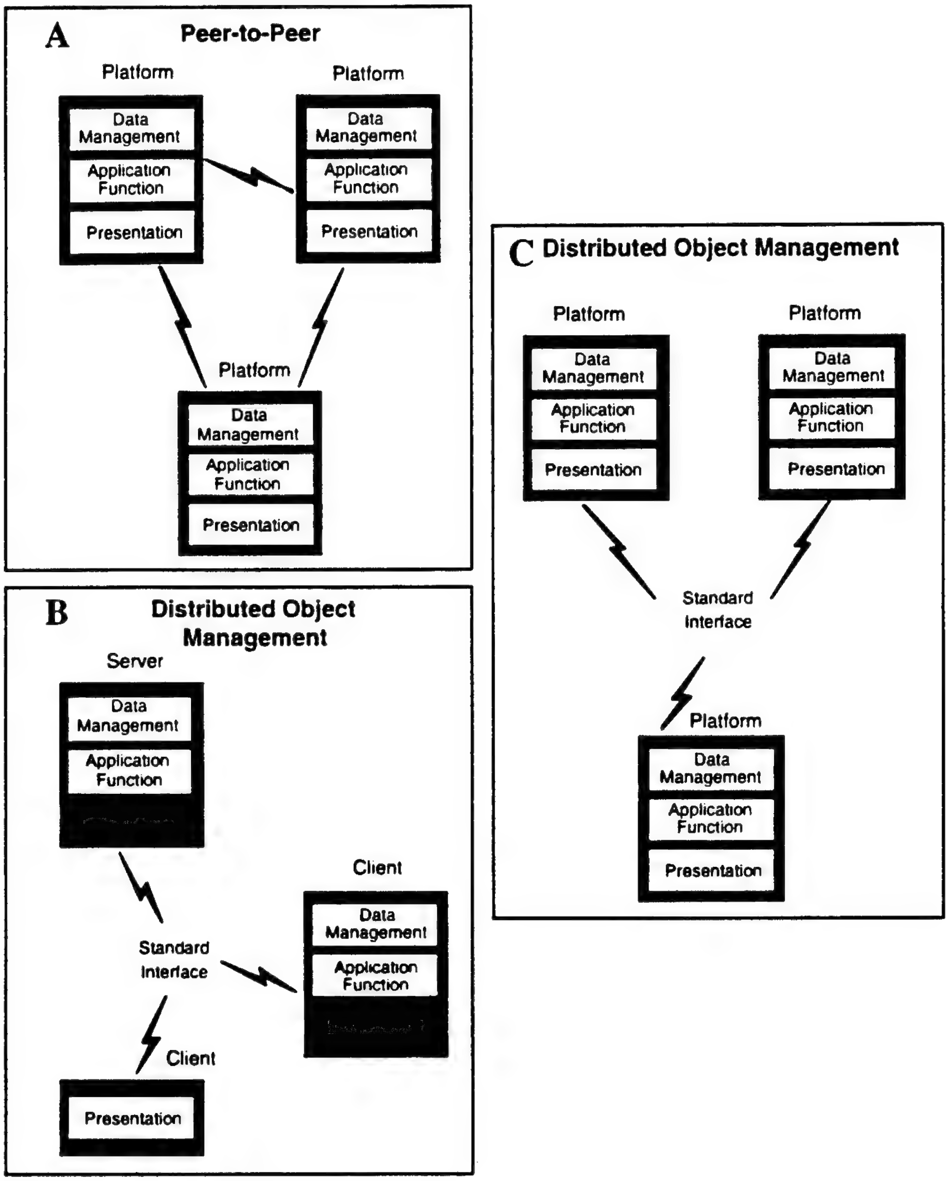


Figure 2-7. Peer-to-Peer and Distributed Object Management Models

The Object Management Group (OMG), a consortium of industry participants working toward object standards, has developed an architecture – the Common Object Request Broker Architecture (CORBA), which specifies the protocol a client application must use to communicate with an Object Request Broker (ORB), which provides services. The ORB specifies how objects can transparently make requests and receive responses. In addition, Microsoft's Object Linking and Embedding (OLE) standard for Windows is an example of an implementation of distributed object management, whereby any OLE-compatible application can work with data from any other OLE-compatible application.

2.3 DATA MANAGEMENT VIEW

The DoD is accomplishing a phased convergence to an open systems environment. This involves the selection of migration systems, defining interim architectures, and performing functional and technical integration. Under the TAFIM, data management services may be provided by a wide range of implementations. Some examples are:

- Mega centers providing functionally oriented corporate databases supporting local and remote data requirements
- Distributed database management systems that support the interactive use of partitioned and partially replicated databases
- File systems provided by operating systems, which may be used by both interactive and batch processing applications.

Data management services include the storage, retrieval, manipulation, backup, restart/recovery, security, and associated functions for text, numeric data, and complex data such as documents, graphics, images, audio, and video. The operating system provides file management services, but they are considered here because many legacy databases exist as one or more files without the services provided by a Database Management System (DBMS).

Major components that provide data management services that are discussed in this section are:

- DBMSs
- Data dictionary/directory systems
- Data security.

These are critical aspects of data management for the following reasons. The DBMS is the most critical component of any data management capability, and a data dictionary/directory system is necessary in conjunction with the DBMS as a tool to aid the administration of the database. Data security is a necessary part of DoD's overall policy for secure information processing.

2.3.1 Database Management Systems

A DBMS provides for the systematic management of data. This data management component provides services and capabilities for defining the data, structuring the data, accessing the data, as well as security and recovery of the data. A DBMS performs the following functions:

- Structures data in a consistent way
- Provides access to the data
- Minimizes duplication
- Allows reorganization, that is, changes in data content, structure, and size
- Supports programming interfaces
- Provides security and control.

A DBMS *must* provide:

- Persistence–The data continues to exist after the application's execution has completed
- Secondary storage management
- Concurrency
- Recovery
- Data definition language/data manipulation language (DDL/DML) – it may be a graphical interface.

2.3.1.1 Database Models

The logical data model that underlies the database characterizes a DBMS. The common logical data models are listed in Figure 2-8. The subsections below discuss each of these database types.

2.3.1.1.1 The Relational Model

A RDBMS structures data into tables that have certain properties:

- Each row in the table is distinct from every other row.
- Each row contains only atomic data; that is, there is no repeating data or such structures as arrays.
- Each column in the relational table defines named data fields or attributes.

Data Model
Relational
Hierarchical
Network
Object-Oriented
Flat File

Figure 2-8. Summary of Data Models

A row of data in a relational database is commonly referred to as a tuple; an example would be a record in a file. An example of a column in a relational table would be a field in a record. A collection of related tables in the relational model makes up a database.

The mathematical theory of relations underlies the relational model – both the organization of data and the languages that manipulate the data. Edgar Codd, then at International Business Machines (IBM), developed the relational model in 1973. It has been popular, in terms of commercial use, since the early 1980s.

2.3.1.1.2 The Hierarchical Model

The hierarchical data model organizes data in a tree structure. There is a hierarchy of parent and child data segments. This structure implies that a record can have repeating information, generally in the child data segments. For example, an organization might store information about an employee, such as name, employee number, department, salary. The organization might also store information about an employee's children, such as name and date of birth. The employee and children data forms a hierarchy, where the employee data represents the parent segment and the children data represents the child segment. If an employee has three children, then there would be three child segments associated with one employee segment. In a hierarchical database the parent-child relationship is one to many. This restricts a child segment to having only one parent segment. Hierarchical DBMSs were popular from the late 1960s, with the introduction of IBM's Information Management System (IMS) DBMS, through the 1970s.

2.3.1.1.3 The Network Model

The popularity of the network data model coincided with the popularity of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. So, the network model permitted the modeling of many-to-many relationships in data. In 1971, the Conference on Data Systems Languages (CODASYL) formally defined the network model. The basic data modeling construct in the network model is the set construct. A set consists of an owner record type, a set name, and a member record type. A member record type can have that role in more than one set, hence the multiparent concept is supported. An owner record type can also be a member or owner in another set. The CODASYL network model is based on mathematical set theory.

2.3.1.1.4 The Object-Oriented Model

An object-oriented DBMS (OODBMS) must be both a DBMS and an object-oriented system. As a DBMS it must provide the capabilities identified above in Section 2.3.1. OODBMSs typically can model tabular data, complex data, hierarchical data, and networks of data. The following are mandatory features an object-oriented system should support:

- **Complex objects** – e.g., objects may be composed of other objects.
- **Object identity** – Each object has a unique identifier external to the data.
- **Encapsulation** – An object consists of data and the programs (or methods) that manipulate it.
- **Types or classes** – A class is a collection of similar objects.
- **Inheritance** – Subclasses inherit data attributes and methods from classes.
- **Overriding with late binding** – The method particular to a subclass can override the method of a class at run time.
- **Extensibility** – e.g., a user may define new objects.
- **Computational completeness** – A general purpose language, such as Ada, C, or C++, is computationally complete. The special-purpose language SQL is not. Most OODBMSs incorporate a general-purpose programming language.

2.3.1.1.5 Flat Files

A flat file system is usually closely associated with a storage access method. An example is IBM's indexed sequential access method (ISAM). The models discussed earlier in this section are logical data models–flat files require the user to work with the physical layout of the data on a storage device. For example, the user must know the exact location of a data item in a record. In addition, flat files do not provide all of the services of a DBMS, such as naming of data, elimination of redundancy, and concurrency control. Further, there is no independence of the data and the application program. The application program must know the physical layout of the data.

2.3.1.2 Distributed DBMSs

A distributed DBMS manages a database that is spread over more than one platform. The database can be based on any of the data models discussed above (except the flat file). The database can be replicated, partitioned, or a combination of both. A replicated database is one in which full or partial copies of the database exist on the different platforms.

A major issue with replication is the method of maintaining consistency between the copies of the database. Some database management systems attempt to do this using complex synchronization algorithms (e.g., "two-phase commit" protocols). Many commercial database vendors are offering a simpler form of replication in which a master copy is updated, then changes are propagated to the database copies by a replication server at a later time.

A partitioned database is one in which part of the database is on one platform and parts are on other platforms. The partitioning of a database can be vertical or horizontal. A vertical partitioning puts some fields and the associated data on one platform and some fields and the associated data on another platform. For example, consider a database with the following fields: employee identification (ID), employee name, department, number of dependents, project assigned, salary rate, tax rate. One vertical partitioning might place employee ID, number of dependents, salary rate, and tax rate on one platform and employee name, department, and project assigned on another platform. A horizontal partitioning might keep all the fields on all the platforms but distribute the records. For example, a database with 100,000 records might put the first 50,000 records on one platform and the second 50,000 records on a second platform.

Whether the distributed database is replicated or partitioned, a single DBMS manages the database. There is a single schema (description of the data in a database in terms of a data model, e.g., relational) for a distributed database. The distribution of the database is generally transparent to the user. The term "distributed DBMS" implies homogeneity.

2.3.1.3 Distributed Heterogeneous DBMSs

A distributed, heterogeneous database system is a set of independent databases, each with its own DBMS, presented to users as a single database and system. "Federated" is used synonymously with "distributed heterogeneous." The heterogeneity refers to differences in data models (e.g., network and relational), DBMSs (e.g., Oracle and Ingres), platforms (e.g., VAX and Sun), or other. The simplest kinds of federated database systems are commonly called gateways. In a gateway, one vendor (e.g., Oracle) provides single-direction access through its DBMS to another database managed by a different vendor's DBMS (e.g., IBM's DB2). The two DBMSs need not share the same data model. For example, many RDBMS vendors provide gateways to hierarchical and network DBMSs.

There are federated database systems both on the market and in research that provide more general access to diverse DBMSs. These systems generally provide a schema integration component to integrate the schemas of the diverse databases and present them to the users as a single database, a query management component to distribute queries to the different DBMSs in the federation, and a transaction management component, to distribute and manage the changes to the various databases in the federation.

2.3.2 Data Dictionary/Directory Systems

The second component providing data management services, the data dictionary/directory system (DD/DS), consists of utilities and systems necessary to catalog, document, manage, and use metadata (data about data). An example of metadata is the following definition: a 6-character long alphanumeric string, for which the first character is a letter of the alphabet and each of the remaining 5 characters is an integer between 0 and 9; the name for the string is employee ID. The DD/DS utilities make use of special files that contain the database schema. (A schema, using metadata, defines the content and structure of a database.) This schema is represented by a set of tables resulting from the compilation of DDL statements. The DD/DS is normally provided as part of a DBMS but is sometimes available from alternate sources. In the management of distributed data, distribution information may also be maintained in the network directory system. In this case, the interface between the DD/DS and the network directory system would be through the API of the network services component on the platform.

In current environments, data dictionaries are usually integrated with the DBMS, and directory systems are typically limited to a single platform. Network directories are used to expand the DD/DS realms. The relationship between the DD/DS and the network directory is an intricate combination of physical and logical sources of data.

2.3.3 Data Administration

DoD Directive (DoDD) 8320.1 defines the data administration program for the DoD. Data administration properly addresses the data architecture, which is outside the scope of the TAFIM. We discuss it briefly here because of areas of overlap. It is concerned with all of the data resources of an enterprise, and as such there are overlaps with data management, which addresses data in databases. Two specific areas of overlap are the repository and database administration, which are discussed briefly below.

2.3.3.1 Repository

A repository is a system that manages all of the data of an enterprise, which includes data and process models and other enterprise information. Hence, the data in a repository is much more extensive than that in a DD/DS, which generally defines only the data making up a database.

2.3.3.2 Database Administration

Data administration and database administration are complementary processes. Data administration is responsible for data, data structure, and integration of data and processes. Database administration, on the other hand, includes the physical design, development, implementation, security, and maintenance of the physical databases. Database administration is responsible for managing and enforcing the enterprise's policies related to individual databases.

2.3.4 Data Security

The third component providing data management services is data security procedures and technology measures that are implemented to prevent unauthorized access, modification, use, and dissemination of data stored or processed by a computer system. Data security also includes data integrity (i.e., preserving the accuracy and validity of the data), and protecting the system from physical harm (including preventative measures and recovery procedures).

Authorization control allows only authorized users to have access to the database at the appropriate level. Guidelines and procedures can be established for accountability, levels of control, and type of control. Authorization control for database systems differs from that in traditional file systems because, in a database system, it is not uncommon for different users to have different rights to the same data. This requirement encompasses the ability to specify subsets of data and to distinguish between groups of users. In addition, decentralized control of authorizations is of particular importance for distributed systems.

Data protection is necessary to prevent unauthorized users from understanding the content of the database. Data encryption, as one of the primary methods for protecting data, is useful for both information stored on disk and for information exchanged on a network.

2.4 COMMUNICATIONS VIEW

The Open Systems Interconnection (OSI) model discussed in the following sections is useful as an aid to understanding the elements of successful network communication; however, it should be understood that the OSI protocols contained in the GOSIP standard are no longer mandated for use by Federal agencies. This change resulted from the emergence of Internet Protocol Suite (IPS) standards as the dominant standards for commercial hardware and software, and the relatively smaller number of OSI-compliant products available. Government agencies are now able to select cost-effective, off-the-shelf networking products that implement open standards, such as those developed by the Internet Engineering Task Force (IETF), International Telecommunications Union, and the International Organization for Standardization (ISO). The OSI protocols have been updated and are contained in the Industry/Government Open System Specification, NIST Publication 500-217.

Communications networks are constructed of end devices (e.g., printers), processing nodes, communication nodes (switching elements), and the linking media that connect them. The communications network provides the means by which information is exchanged. Forms of information include data, imagery, voice, and video. Automated information systems (AISs) accept and process information using digital data formats rather than analog formats. Therefore, TAFIM communications concepts and guidance will focus on digital networks and digital services. Integrated multimedia services are included.

The communications view describes the architecture of DoD communications with respect to its geography (Section 2.4.1), discusses the OSI reference model, and describes a general framework intended to permit effective system analysis and planning for DoD (Section 2.4.2).

2.4.1 DoD Communications Infrastructure

The communications infrastructure in the DoD will contain three transport components, local, regional/metropolitan, and global, as shown in Figure 2-9. The names of the transport components are based on their respective geographic extent, but there is also a hierarchical relationship among them.

The transport components correspond to a network management structure in which management and control of network resources are distributed across the different levels.

The local components relate to assets that are located relatively close together geographically. This component contains sustaining base communications assets for the fixed environment and tactical communications assets in the deployed environment. LANs, to which the majority of end devices will be connected, are included in this component. Standard interfaces will facilitate portability, flexibility, and interoperability of LANs and end devices.

Regional and metropolitan area networks (MAN) are geographically dispersed over a large area. A regional or metropolitan network could connect local components at several sustaining bases in the fixed environment or connect theater tactical assets in the deployed environment. In most cases, regional and metropolitan networks are used to connect local networks. However, shared databases, regional processing platforms, and network management centers may connect directly or through a LAN. Standard interfaces will be provided to connect local networks and end devices.

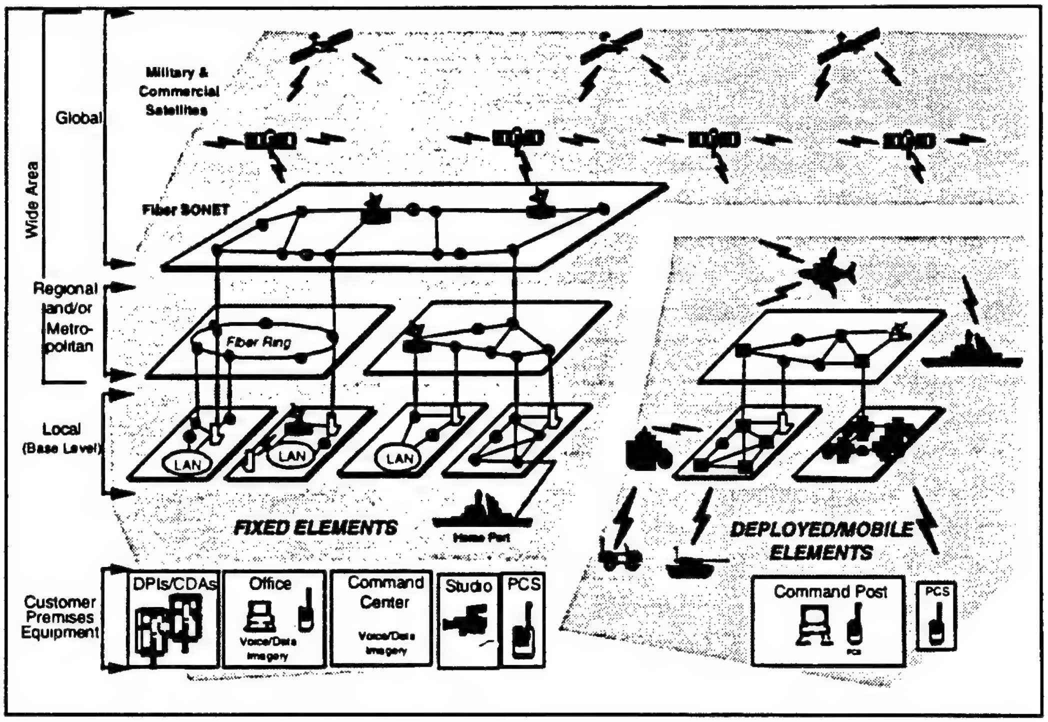


Figure 2-9. Communications Infrastructure

Global or wide area networks (WAN) are located throughout the world, providing connectivity for regional and metropolitan networks in the fixed and deployed environment. In addition, deployed mobile assets, shared databases, and central processing centers can connect directly to the global network as required. Standard interfaces will be provided to connect regional and metropolitan networks and end devices.

The network that will support all DoD data transport requirements is the Defense Information Systems Network (DISN), authorized under the Chairman of the Joint Chiefs of Staff (CJCS) Memorandum of Policy (MOP) 70, dated 5 February 1992. DISN is intended to support National Defense Command, Control, Communications, and Intelligence (C3I) decision support requirements; Corporate Information Management (CIM) functional business areas; and Defense Information Infrastructure (DII) data processing and information transfer services. The DISN will support the DoD's WAN on the global scale. DISN responsibility will extend to the local component in the future.

2.4.2 Communications Models

The geographically divided infrastructure described in Section 2.4.1 forms the foundation for an overall communications framework. These geographic divisions permit the separate application of different management responsibilities, planning efforts, operational functions, and enabling technologies to be applied within each area. Hardware and software components and services fitted to the framework form the complete model.

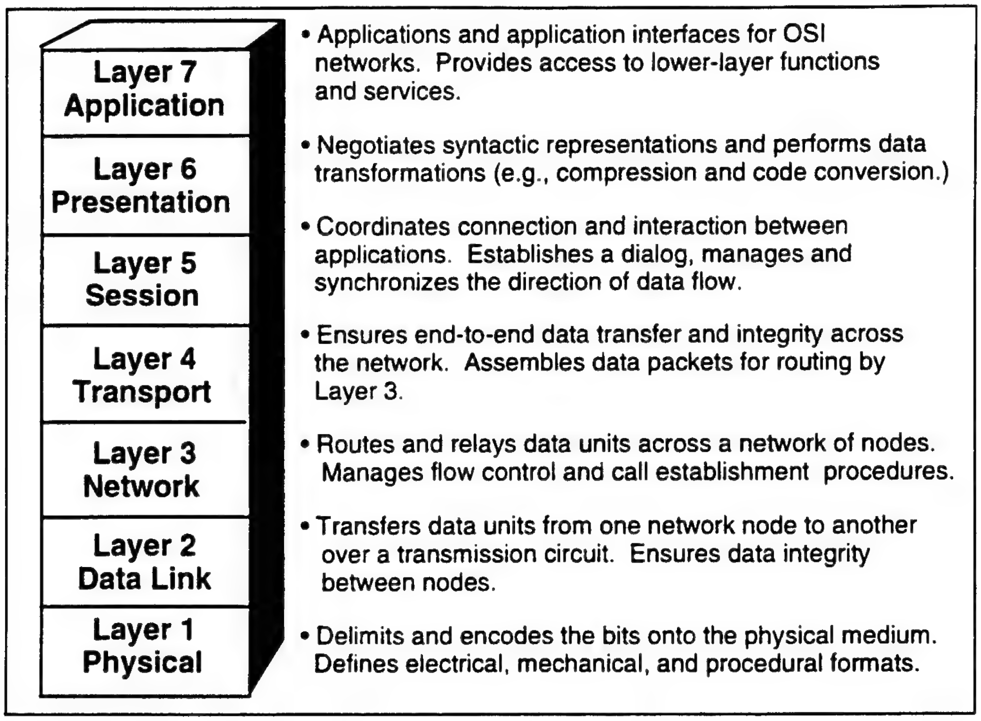
The following sections describe the OSI reference model and a grouping of the OSI layers that facilitates discussion of interoperability issues.

2.4.2.1 The OSI Reference Model

The OSI reference model, portrayed in Figure 2-10, is the model used for data communications in the TAFIM. Although DoD is no longer advocating the full use of OSI protocols, the OSI reference model is a valuable tool for conceptualizing networking requirements and solutions. Each of the seven layers in the model represents one or more services or protocols (a set of rules governing communications between systems), which define the functional operation of the communications between user and network elements.

Each layer provides services for the layer above it. This model aims at establishing open systems operation and implies standards-based implementation. It strives to permit different systems to accomplish complete interoperability and quality of operation throughout the network.

The seven layers of the OSI model are structured to facilitate independent development within each layer and to provide for changes independent of other layers. Stable international standard protocols in conformance with the OSI reference model layer definitions have been published by various standards organizations. Support and mission-area applications, as defined in the DoD Technical Reference Model, are above the OSI Reference Model protocol stack and use its services via the applications layer.



- Applications and application interfaces for OSI networks. Provides access to lower-layer functions and services.
- Negotiates syntactic representations and performs data transformations (e.g., compression and code conversion.)
- Coordinates connection and interaction between applications. Establishes a dialog, manages and synchronizes the direction of data flow.
- Ensures end-to-end data transfer and integrity across the network. Assembles data packets for routing by Layer 3.
- Routes and relays data units across a network of nodes. Manages flow control and call establishment procedures.
- Transfers data units from one network node to another over a transmission circuit. Ensures data integrity between nodes.
- Delimits and encodes the bits onto the physical medium. Defines electrical, mechanical, and procedural formats.

Figure 2-10. Open Systems Interconnection Model

2.4.2.2 Communications Framework

A communications system based on the OSI reference model includes services of all the layers described in the previous section plus the physical transmission media and the support and mission-area applications defined in Volume 2, *Technical Reference Model*. These elements may be grouped into architectural levels that represent major functional capabilities, such as switching and routing, data transfer, and the performance of applications.

These architectural levels are:

- The Transmission Level (below the OSI) provides all of the physical and electronic capabilities, which establish a transmission path between functional system elements (wires, leased circuits, interconnects, etc.).
- The Network Switching Level (OSI layers 1 through 3) establishes connectivity through the network elements to support the routing and control of traffic (switches, controllers, network software, etc.).
- The Data Exchange Level (OSI layers 4 through 7) accomplishes the transfer of information after the network has been established (end-to-end, user-to-user transfer) involving more capable processing elements (hosts, workstations, servers, etc.).
- The Applications Program Level (above the OSI) includes the support and mission-area applications (non-management application programs).

The communications framework is defined to consist of the three geographical components of the DoD communications infrastructure (local, regional, and global) and the four architectural levels (transmission, network switching, data exchange, and application program), and is depicted in Figure 2-11. Communications services are performed at one or more of these architectural levels within the geographical components.

Figure 2-11 shows computing elements (operating at the applications program level) with supporting data exchange elements, linked with each other through various switching elements (operating at the network level), each located within its respective geographical component. Figure 2-11 also identifies the relationship of the Technical Reference Model to the communication architecture.

2.4.2.3 Allocation of Services to Components

The DoD communications infrastructure consists of the local, regional, and global transport components. The services allocated to these components are identical to the services of the application program, data exchange, network switching, or transmission architectural levels that

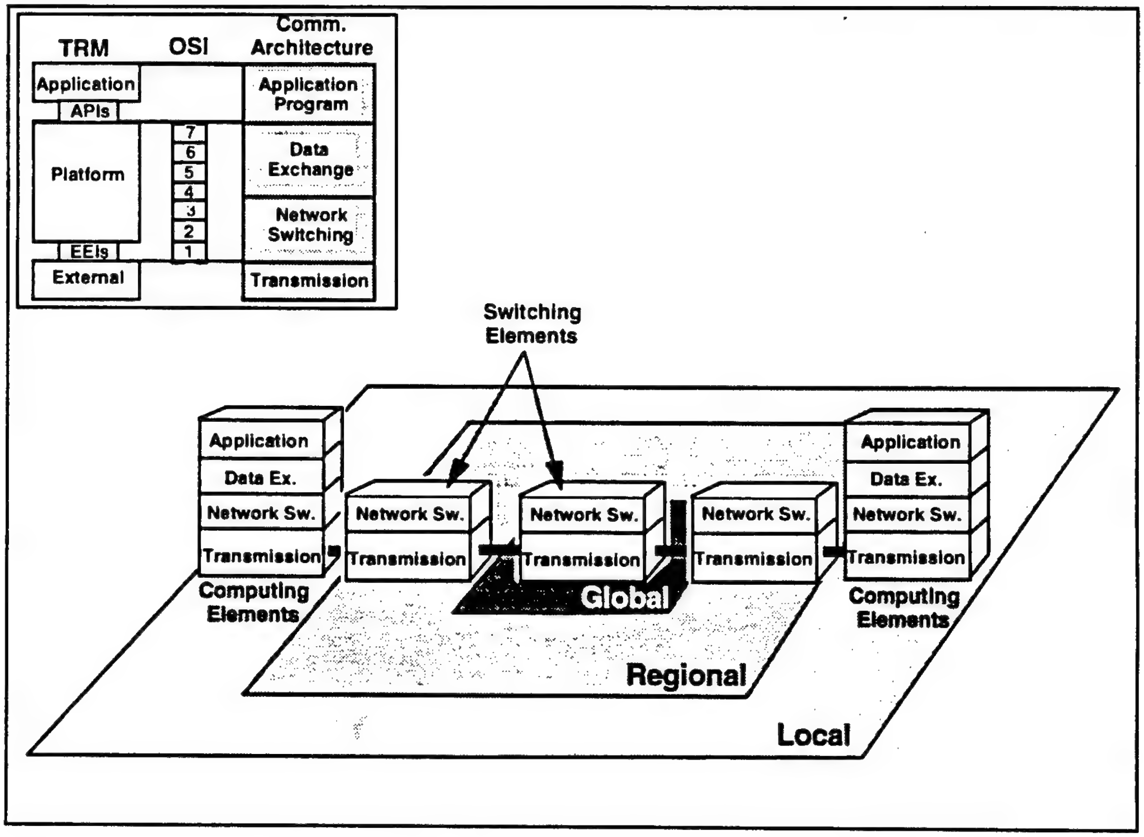


Figure 2-11. Communications Framework

apply to a component. Data exchange and network switching level services are identical to the services of the corresponding OSI reference model layers. Typically, only network switching and transmission services are allocated to the regional and global components, which consist of communications nodes and transmission media. All services may be performed in the local component, which includes end devices, processing nodes, communications nodes, and linking media. Transmission, switching, transport, and applications are all performed in this component.

2.5 SECURITY VIEW

The business of the DoD requires the controlled use of information. Security protection of DoD information systems is discussed in Volume 6 of the TAFIM, *DoD Goal Security Architecture (DGSA)*. The purpose of this section is to provide a brief overview of Volume 6 with a focus on security protection implemented in the information system components. Doctrinal mechanisms, such as physical and personnel security procedures and policy, are discussed in Volume 6 but omitted here.

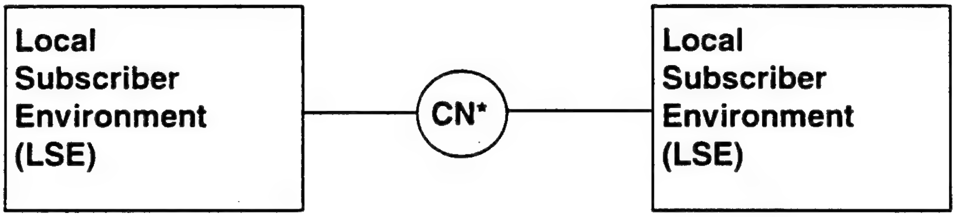
Figure 2-12 depicts an abstract view of an information system architecture, which emphasizes the fact that an information system from the security perspective is either part of a local subscriber environment (LSE) or a communications network (CN). An LSE may be either fixed or mobile. The LSEs by definition are under the control of the using organization. In an open system distributed computing implementation, secure and nonsecure LSEs will interoperate.

2.5.1 Basic Concepts

This section presents basic concepts required for an understanding of information system security within DoD.

2.5.1.1 Information Domains

The concept of an *information domain* provides the basis for discussing security protection requirements. An information domain is defined as a set of users, their information objects, and a security policy. An information domain security policy is the statement of the criteria for membership in the information domain and the required protection of the information objects.


Local Subscriber Environment (LSE)
CN*
Local Subscriber Environment (LSE)

*CN = Communications Network

Figure 2-12. Abstract Security Architecture View

The missions of most DoD organizations require that their members operate in more than one information domain. The diversity of mission activities and the variation in perception of threats to the security of information will result in different information domains within one mission security policy. A specific mission may use several information domains, each with its own distinct information domain security policy.

There must be no security-relevant distinction made among the information objects in an information domain. Members of an information domain may have different security-related attributes. For example, some members might have only read permission for information objects in an information domain, while other members might have both read and write permissions.

Since all information objects in an information domain have the same security-relevant attributes, a user who has read and write permissions in an information domain has those permissions for every information object in the information domain. The term "information object" refers to any type of information.

Information domains are not bounded by information systems or even networks of systems. The security mechanisms implemented in information system components may be evaluated for their ability to meet the information domain security policies.

2.5.1.2 Strict Isolation

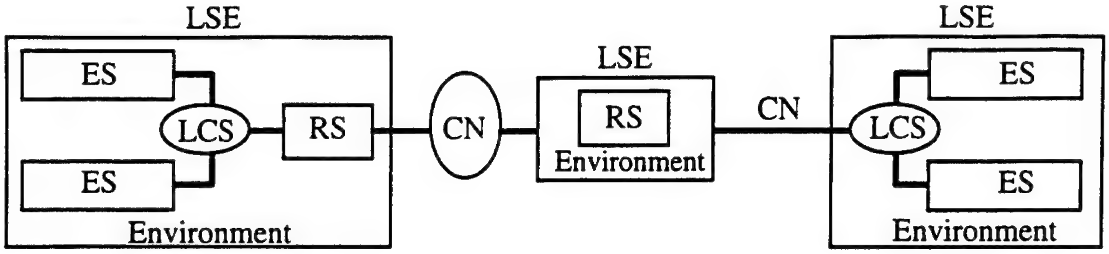
The strategy of "strict isolation" is used to isolate one information domain from another. Information objects can be transferred between two information domains only in accordance with established rules, conditions, and procedures expressed in the security policy of each information domain. Multidomain information objects may be defined for display or printing. A multidomain information object is a defined collection of information objects from multiple information domains.

2.5.1.3 Absolute Protection

The concept of "absolute protection" is used to provide a framework for achieving uniformity of protection in all information systems supporting a particular information domain. It directs attention to the problems created by the interconnection of LSEs that provide disparate strengths of security protection. This possibility is likely because open systems will consist of an unbounded number of unknown heterogeneous LSEs that must be able to interoperate. Analysis related to minimum assurance requirements will ensure that the concept of absolute protection will be achieved for each information domain across LSEs.

2.5.2 Security Generic Architecture View

Figure 2-13 shows the generic architectural view used in Volume 6 to discuss the allocation of security services and the implementation of security mechanisms. This view identifies the architectural components within a LSE. The LSEs are connected by CNs. The LSEs include end systems, relay systems, and local communications systems (LCSs), described below.



Key

CN - communications network

ES - end system

LCS - local communications system

LSE - local subscriber environment

RS - relay system

Figure 2-13. Generic Security Architecture View

- **Relay system** – The component of an LSE, the functionality of which is limited to information transfer and is only indirectly accessible by users (e.g., router, switch, multiplexer, message transfer agent). It may have functionality similar to an end system, but an end user does not use it directly. Note that relay system functions may be provided in an end system.
- **Local communication system** – A network that provides communications capabilities between LSEs or within a LSE with all of the components under control of a LSE.
- **Communication network** – A network that provides inter-LSE communications capabilities, but is not controlled by LSEs (e.g., commercial carriers).

The end system and the relay system are viewed as requiring the same types of security protection. For this reason, a discussion of security protection in an end system generally also applies to a relay system. The security protections in an end system could occur in both the hardware and software.

2.5.3 Security Services Allocation

Security protection of an information system is provided by mechanisms implemented in the hardware and software of the system and by the use of doctrinal mechanisms. The mechanisms implemented in the system hardware and software are concentrated in the end system or relay system. This focus for security protection is based on the open system, distributed computing approach for DoD information systems. This implies use of commercial common carriers and DoD-owned common-user communications systems as the CN provider between LSEs. Thus, for operation of end systems in a distributed environment, a greater degree of security protection can be assured from implementation of mechanisms in the end system or relay system.

However, CNs should satisfy the availability service to promote satisfaction of appropriate security protection for the information system. This means that CNs must provide an agreed level of responsiveness, continuity of service, and resistance to accidental and intentional threats to the communications service availability.

End systems may not need to interoperate with others, but may need to accommodate multiple security domains processing simultaneously.

Implementing the necessary security protection in the end system occurs in three system service areas. They are operating system services, network services, and system management services.

Most of the implementation of security protection is expected to occur in software. The hardware is expected to protect the integrity of the end system software. Hardware security mechanisms include protection against tampering, undesired emanations, and cryptography.

2.5.3.1 Operating System Services

A "security context" is defined as a controlled process space subject to an information domain security policy. The security context is therefore analogous to a common operating system notion of user process space. Isolation of security contexts is required. Security contexts are required for all applications (e.g., end user and security management applications). The focus is on strict isolation of information domains, management of end system resources, and controlled sharing and transfer of information among information domains. Security-critical functions are isolated into relatively small modules that are related in well-defined ways.

The operating system "separation kernel" will maintain the required isolation. The separation kernel will use the protection features of the end system hardware (e.g., processor state register, memory mapping registers) to maintain strict separation among security contexts by creating separate address spaces for each of them. Untrusted software will use end system resources only by invoking security-critical functions through the separation kernel. Security-critical functions perform inter-security context (i.e., inter-information domain) operations. Most of the security-critical functions are the low-level functions of traditional operating systems.

2.5.3.2 Network Services

Two basic classes of communications are envisioned for which distributed security contexts may need to be established. These are interactive and staged (store and forward) communications.

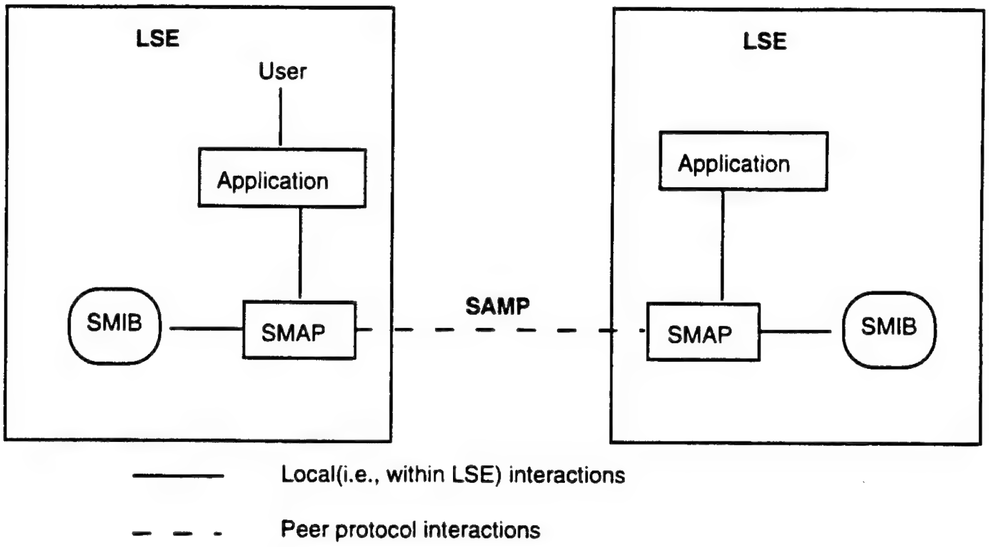
The concept of a "security association" forms an interactive distributed security context. A security association is defined as the totality of communication and security mechanisms and functions to extend the protections required by an information domain security policy within an end system to information in transfer between multiple end systems. The security association is an extension or expansion of an OSI application layer association. An application layer association is composed of appropriate application layer functions and protocols plus all of the underlying communications functions and protocols at other layers of the OSI model. Multiple security protocols may be included in a single security association to provide for a combination of security services. However, a security association can only be established within the same information domain; inter-information- domain security associations are not allowed.

For staged delivery communications (e.g., e-mail), use will be made of an encapsulation technique (termed "wrapping process") to convey the necessary security attributes with the data being transferred as part of the network services. The wrapped security attributes are intended to permit the receiving end system to establish the necessary security context for processing the transferred data. If the wrapping process cannot provide all the necessary security protection, interactive security contexts between end systems will have to be used to ensure the secure staged transfer of information.

2.5.3.3 System Security Management Services

Security management is a particular instance of general information system management functions as discussed in Volume 2. Information system security management services are concerned with the installation, maintenance, and enforcement of information domain and information system security policy rules in the information system intended to provide these security services. In particular, the security management function controls information needed by operating system services within the end system security architecture. In addition to these core services, security management requires event handling, auditing, and recovery. Standardization of security management functions, data structures, and protocols will enable interoperation of security management application processes (SMAPs) across many platforms in support of distributed security management. Areas for security management standardization are described in Volume 6, *DoD Goal Security Architecture (DGSA).*

SMAPs, using information in the information base, will be used to establish the required security contexts for interactive communications among distributed platforms operating in various information domains simultaneously. System SMAPs will also be used to provide the security protection of store-and-forward communications in which the requisite security contexts cannot be handled within the message. The end system will establish a security association by using a SMAP, a security association management protocol (SAMP), and information in the security management information base (SMIB). Figure 2-14 shows the general relationship of the processes and protocols involved in establishing a security association for interactive communications among distributed end systems.


LSE
User
Application
SMIB
SMAP
SAMP
LSE
Application
SMAP
SMIB
Local(i.e., within LSE) interactions
Peer protocol interactions

Key

LSE - Local subscriber environment
SAMP - Security association management protocol
SMAP - Security management application process
SMIB - Security management information base

Figure 2-14. Architectural Components Involved in Establishing a Security Association for Interactive Communications

3.0 DESIGN GUIDANCE

The architectural concepts discussed in Section 2 are needed to realize the long-term DoD IM vision of an open distributed computing environment. In the near term, new information systems will be engineered and integrated into an environment that includes legacy and migration systems. Many legacy and migration systems use non-standard and proprietary approaches. This complicates the integration of new systems designed for an open systems environment. This chapter focuses on near-term application of the architecture concepts. It includes guidance for integrating open systems, legacy systems, and migration environments. The guidance presented in this section will evolve over time based on the availability and maturity of technology. The guidance in this section supplements the concepts in Volume 2.

3.1 GUIDANCE FOR DESIGNING ARCHITECTURES

An architecture is a set of components and a specification of how these components are connected to meet the overall requirements of an information system. The components of an architecture provide implementations of the reference model services relevant to a specific system. The following are guidelines for designing specific architectures given the Technical Reference Model and the model of information system architecture:

- An architecture will contain components to implement only those reference model services that it requires.

- Components may implement one, more than one, or only part of a service identified in the reference model.

- The components should conform to the profile standards that are relevant to the services they do implement.

The following is a general procedure for designing a specific architecture given the guidelines above:

- Perform requirements analysis

- Make service allocations

- Select components

- Evaluate.

3.2 COMPUTING MODELS

This section presents guidance on the computing models discussed in Section 2.2.

3.2.1 Client/Server Model Design Analysis and Guidance

The objective computing environment is a distributed computing environment based on open systems principles and public standards (e.g., Federal Information Processing Standards [FIPS] and ISO). In this environment, services will be provided by servers distributed to processing nodes throughout a network. Services will be distributed based on attention to survivability, efficiency, and functional requirements. As DoD migrates to a distributed computing environment, many different types of client/server capabilities will be established. These will support, among other things, the initial implementation of subject area databases, access to data managed by migration systems from open system platforms, and interactive applications that are distributed to a POSIX platform and accessed via a standards-compliant graphical user interface (GUI). The following are some guidelines related to the client/server model:

- Provide client processes with a high-level interface with as few details about the underlying communication and processing details as possible. An example interface is that provided by the CORBA. Design the client/server mechanism so that the location of the server can change without impacting the client application.

- Isolate the client application from the details of the interprocess application. This would allow details of the communication mechanism to change without affecting the client application. This would also allow requests for services to be provided by both local and remote servers, and the client would be insulated from the details.

- Design subject area databases to provide users and client processes with the capability to query data without knowing where or how the data is physically stored. Use distributed processes to provide record level access to mainframe migration databases. Until products are readily available that comply with an open standard, the preferred method is through a de facto standards-based implementation of the client/server model. This will require implementation of a de facto standard on the appropriate mainframe platform. End-user workstations that require this type of access will also require a de facto standard implementation. This will give distributed users and client processes access to both query and update data managed by migration mainframe databases.

- Use other implementations of the client/server model achievable today, including access to file and print servers provided through a network file service. This can provide users with remote access to files and network printers. These services should be provided in such a way that there is a migration path to open standards when they become available.

To summarize, the client/server model provides a flexible framework for designing and maintaining distributed processing applications. New application-enabling technology, such as computer-aided software engineering (CASE) and GUIs, focus on this model. Some general advantages of the model are:

- Readily supports the open systems concepts of portability, interoperability, scalability, modularity, and flexibility
- Allows for the continued use of existing capital investments such as PCs, LANs, minicomputers, and mainframes
- Enables data sharing among many different applications
- Accommodates special function hardware or software that does not have to be duplicated
- Allows data and processing to be distributed to the appropriate organizational level
- Supports centralized control and security of data
- Supports survivability through the management and distribution of redundant data and processes
- Supports consistent user interfaces across applications.

3.2.1.1 Guidance on the Multitiered Architecture

Industry evolved from single-tiered architectures on mainframes to two-tiered architectures (often still on mainframes) because there was a recognized need to separate data from applications. In the single-tier architecture, each application had its own data in its own files and there was little, if any, opportunity for application A to share application B's data. Once the data was separated from the applications, which has often happened in actual implementations of the two-tiered client/server architecture, the data became a resource that could be used across multiple applications. Inconsistent data sets were eliminated (in concept at least), concurrent access to data was allowed, integrity constraints on data were supported, and data was protected.

The multitiered approach is an extension of the two-tiered approach. By now separating the user interface (i.e., the presentation layer) from the rest of the application (remember, data has already been separated out), the functional code of the application can be turned into the elements of a reusable library of functional routines. Furthermore, those routines can begin to be executed on networked platforms (possibly remote from the user platform); the elements of the presentation layer can also be turned into a library of reusable elements, and the user interface can be replaced with a new interface without having to rewrite the entire application.

The multitiered approach will allow migration of legacy systems to modular systems and taking advantage of the benefits mentioned above. However, the separation of the application logic and presentation layers may exclude certain COTS systems that would otherwise offer significant benefits – in particular, there are COTS products (two-tier) that offer the benefit of porting their clients to multiple platforms with a simple recompilation. There are also products (again two-tier) that offer very powerful data access and manipulation capabilities across multiple data servers (like cross-database joins) that would otherwise have to be coded and maintained as part of the "functional" code.

New system and migration system architectures should carefully consider the relative advantages of specific COTS products as well as the two-tiered and multitiered approaches based on the specific system's requirements. No single solution will meet the needs of all DoD systems.

3.2.2 Guidance for Other Computing Models

The compelling advantages of the client/server model must be weighed against many potential disadvantages. The requirements of some environments cannot withstand the potential disadvantages of a migration to client/server in the near term. When considering migration from a centralized or mainframe environment, the following disadvantages of the client/server model should be considered:

- Client/server computing is heavily dependent upon the reliability and performance characteristics of the network.

- Security and data integrity requirements are more complicated than when processing is performed on a single (e.g., mainframe) platform. Access to servers and services must be limited to authorized clients. Each client also must be able to select a specific server and be assured that only this server gets access to its data during callbacks. The sender of messages must be assured that messages are neither read nor distorted by other parties. A secure message requires an authentication service and an encryption technique. The message protocol must be able to support the needed security services as determined at runtime.

- Client/server implementations usually entail the integration of a more diverse set of products (than for mainframe-based implementations), increasing the integration effort, complexity, and risk.

- System management, administration, performance monitoring, fault isolation, and correction are more difficult. This is due principally to a lack of tools, a more complex environment, and the interaction of diverse components.

- Development and maintenance staffs that have been working in mainframe environments often do not possess the necessary skills to implement and maintain client/server systems.

- The expected cost saving through the use of low-cost commodity processors may be offset by increased administration costs and the need for highly qualified support personnel.

- Initial client/server implementations often take longer and cost more than expected due to lack of familiarity with the development process and tools.

- It is difficult to replicate problems at a vendor or other central support site. Client/server systems are often a combination of COTS products, protocols, local configuration parameters, and developed software that result in a system with many unique attributes.

- The security implications are not as well understood in comparison with those for mainframe solutions.

When several of these disadvantages are concerns, a client/server implementation may not be the best solution. A mainframe-based approach such as the master-slave or three-tiered model may be more appropriate.

With respect to other models, the following considerations should be addressed. The peer-to-peer approach is often considered a superior alternative to the client/server approach, providing client/server capabilities with the advantage that an application can be processed on any computer in the network – wherever the computing resources are available. However, there are technical challenges associated with the peer-to-peer model that have not been overcome to date, and very few implementations of this model are commercially available. The major value of this type of design approach is that it enhances system availability. This approach could be used to provide system availability for implementations of other computing models. It could provide a hot standby for a centralized system or hot standbys for servers in a client/server implementation. Some types of servers that would benefit from redundancy are naming, communications, and database.

The distributed object management model can be considered as a special case for the client/server or peer-to-peer model, and some consider it superior to the client/server model because of its attempt to provide a cleaner, simpler interface between systems. This model significantly reduces the number of interfaces required among interoperating platforms. It requires only one interface for all platforms as opposed to one interface for each pair of platforms. This is a significant advantage. In addition, once a system has sent a message to another system, the sending system is not required to cease all processing while awaiting a response.

3.3 DATA MANAGEMENT

The near-term goals for improving data management in the DoD are focused on consolidation and interoperation. Consolidation is being carried out primarily in the context of existing applications, resulting in the need to merge databases of similar applications into a single migration system. The surviving system will not necessarily be upgraded to meet target architectural standards. As a result, many existing data management technologies will still be in use at the end of the near-term period. Longer term, the strategic combination of data management components and the standards in Volume 2 of the TAFIM will allow DoD to evolve to an open systems environment. This can be achieved through the identification of flexible data management components and the implementation of systems intended to enhance DoD-wide interoperability.

3.3.1 Guidance on DBMSs

The alternative to a DBMS is a flat file system. The primary advantages of a DBMS are data independence and controlled redundancy. In addition, DBMSs provide capabilities for defining a database through a schema, querying and manipulating the data, concurrency control, and systematic backup and recovery. Flat file systems provide none of these capabilities. A DBMS is preferable to a flat file system.

Flat files might be chosen over DBMSs in the rare event that the application and the file require a very high level of performance (a flat file system does not carry the overhead burden of a DBMS) and very little maintenance.

3.3.1.1 Guidance on Database Models

The RDBMS has become the DBMS of choice over the hierarchical and network model DBMSs. There are several reasons for this. Because of the complexity of the structures involved in the hierarchical and network models, the manner of accessing the data, and the implementation of the structures, both the hierarchical and network models are considered significantly more difficult to manage than the relational model. With the hierarchical and network DBMSs, the application programmer must specify the navigation path for reaching data, e.g., the complete hierarchical path for reaching a data item, as opposed to just the attribute name in the case of an RDBMS. In addition, physical pointers are used in hierarchical and network DBMSs to represent the parent-child and owner-member relationships, respectively. These physical pointers present a difficult challenge in pointer maintenance. The tabular representation of data in the relational model and the relational algebra and calculus languages used to interact with relational databases are considered much simpler than network and hierarchical systems, and application programming and maintenance are much easier. In addition, the performance problems of the early RDBMSs have been overcome, and it has been shown that data that can be modeled as hierarchies and networks can also be modeled as relations. Therefore the RDBMS is recommended over the network and hierarchical DBMSs.

Many experts say we are at the beginning of a new generation of DBMSs – the object-oriented and extended relational. An extended relational DBMS is a relational DBMS with some object-oriented features, generally class inheritance, complex data, or large objects (such as text and graphics). RDBMSs are excellent for conventional or business data, such as personnel and payroll, but not for non-conventional data and applications. Examples of applications requiring non-conventional data are CASE, computer-aided design (CAD), computer-aided manufacturing (CAM), and expert systems. These new applications require multiple data types and the ability to represent complex relationships among the data. In addition, the new applications require modeling power, long and design transactions, and version and configuration management. OODBMSs are being designed and manufactured to meet these needs. A criticism of OODBMSs is that they are reminiscent of network DBMSs with their pointers to implement relationships. However, OODBMSs use logical, not physical pointers, and the difficulty of pointer maintenance present in network DBMSs does not apply to OODBMSs. Another criticism of OODBMSs is that there is no formal theory behind the model, as with the relational model.

However, there are formalisms involved, such as generalization (class/subclass/inheritance), aggregation (objects composed of other objects), and object identity. Standards are not final in this area, and as such the OODBMS presents some risk (e.g., in portability).

OODBMSs fill a gap in the data management area with respect to non-conventional applications that are becoming more prevalent. Most of the commercial OODBMS products are oriented to client/server environments. Procurement of an OODBMS is recommended when the need to support non-conventional applications is present.

3.3.1.2 Guidance on Distributed DBMSs

Replicated databases are recommended when survivability, availability, low transmission cost, and quick response time are important. Replicated databases enhance survivability because if one copy is destroyed in a disaster, other copies are available at other locations. Availability is enhanced for similar reasons. Replicated copies can be located close to the users, so transmission costs are less. For the same reason, response time should be less.

A disadvantage of fully synchronized replicated databases is that updates are expensive because of the need to maintain complete consistency between copies of the database. This form of replication can also result in the inability to update the database if one of the copies is unavailable due to network or system problems. Therefore this form of replication is not recommended when there are frequent database updates. The use of delayed replication servers is recommended except in the rare cases where absolute consistency is required.

A disadvantage of replicated databases is that updates are expensive because of the need to synchronize the updates of the copies of the database. Therefore, this distributed DBMS approach is not recommended when there are frequent database updates.

Partitioned databases are recommended when: 1) there is a high locality of reference – data at a site is used most by local users and infrequently by remote users; 2) retrieval costs are a concern – these costs are lower; and 3) update costs are a concern – these costs are also lower.

3.3.1.3 Guidance on Distributed Heterogeneous DBMSs

Many DBMSs provide gateways to databases managed by other DBMSs. Gateways are recommended when an organization has standardized on one DBMS, but there is other data, either legacy or in another organization, that the organization needs to access. Gateways are generally limited – that is from one DBMS to one or two other DBMSs without a general solution to federating databases. Gateways are usually an option when procuring a DBMS (e.g., a gateway to DB2 when Sybase RDBMS is purchased). Gateways are recommended when there is a specific need to access a second DBMS using the organization's standard DBMS.

Federated database systems are usually sold by third parties for use in integrating databases managed by different vendor DBMSs. For example, a federated database product might be sold that integrates relational, hierarchical, network, and object-oriented DBMSs. These products attempt to present a general solution for integrating data. Typically, a common data model (e.g., entity-relationship, object-oriented, or relational) is used to develop schemas of the shared data from all the databases. A user sees schemas or views in this common data model and accesses any participating database using the common data language that is provided. In this approach a user is not burdened with having to know the data models and languages of all the participating DBMSs. Federated database systems are an acknowledgment that different autonomous groups often make different decisions on which data model (e.g., relational versus object-oriented) or DBMS to use. Yet the results of the different choices may well have to interoperate. This applies to integrating legacy and migration databases, as well as the integration of different open system databases. Federated database systems are recommended for the interoperability of autonomous database systems – legacy, migration, and open.

3.3.2 Guidance on Data Dictionary/Directory Systems

All DBMSs procured should include an integrated DD/DS.

3.3.3 Guidance on Data Administration

The 8320 series of DoDD provides detailed guidance on data administration.

3.3.4 Guidance on Data Security

Commercially available data management components typically provide integrity and availability services. The integrity services are used to maintain internal database consistency, while availability services control concurrent access to database resources. Some degree of identity-based confidentiality protection is also provided by being able to specify the data management commands that certain users are allowed to execute with respect to certain database objects. To protect the confidentiality of classified or unclassified-but-sensitive data, use of a trusted database management system may be recommended.

DBMSs that have undergone the National Computer Security Center (NCSC) evaluation process are recommended for secure environments. It is important to note that the NCSC evaluates DBMSs based on their ability to enforce a defined confidentiality policy. Enforcement of an integrity and availability policy is not part of the current evaluation requirements, although the National Institute of Standards and Technology (NIST) test suite for compliance with the American National Standards Institute (ANSI) SQL standard does test for the support for those features that are part of the current ANSI SQL standard. Since the evaluation of trusted DBMSs is a new procedure, additional guidance in this area may be forthcoming.

In the near term, several options exist: Treat the database management system as an untrusted component operating on a trusted operating system; begin utilization of relational database management systems that are in the evaluation process, while supplementing systems with other controls to compensate for the low assurance rating; or begin using a combination of trusted database management system and trusted operating system capabilities to achieve higher assurance for label-based confidentiality policy. The first approach allows the application to use the full suite of commercial software but does not provide support for needed security functionality. The other two approaches provide confidentiality support to various levels of assurance but may require a sacrifice of application functionality because the full suite of commercial software may not be compatible with the selected trusted products.

Use of client/server architectures in a distributed configuration for data management functions is highly appropriate but may not be fully supported by the current suite of trusted database management system products. In addition, these systems provide limited or no support for distributed data management functions with enforcement of confidentiality. If the data management capability does support operation in a distributed configuration, the other non-data-management components must also be considered. This is required because the point of exposure to security vulnerability involves each system component and the communication system, since requests and responses must be protected as they travel through the distributed system.

3.3.5 Transition Components

In the environment of the future, there will be a DoD information architecture with standardized components and local nonstandard systems. The data architecture will include the DoD data model, standard database structures, standard data elements, and procedures. The current environment of legacy systems will need to migrate to the new environment. In transitioning to this new environment, many of the data management components discussed in Section 2.2 will be used. In the near future, RDBMSs will begin to proliferate. Database gateways and federated systems will exist to link the RDBMSs with legacy data in flat files and hierarchical and network databases. Some OODBMSs will begin to appear, and federated systems will also link them with the RDBMSs and legacy systems. Distributed DBMSs will be used to provide survivability, availability, and the placement of data closer to the users.

3.4 COMMUNICATIONS

The DISN will evolve to become the common, worldwide communications infrastructure. It will provide integrated data, voice, video, and imagery services with connectivity for command and control systems, intelligence systems, and business systems. In the future, DISN will provide consolidated communications services that efficiently satisfy DoD connectivity requirements.

The DISN will provide or facilitate the following capabilities:

- User logon to any number of computers from any number of locations
- Communications at all levels of classification
- Support to real world events as users change locations and network nodes and end devices change locations
- Dynamic network management to ensure that essential communications receive priority, that the network adjusts to the addition or deletion of communications nodes and links, and that messages are routed to intended recipients regardless of their actual location

- Connectivity or gateways to other federal communications networks and commercial networks

- Mobile communications on land, in the air, or at sea that will incorporate wireless service to extend connectivity to mobile users

- Military-unique requirements, including precedence and preemption

- Linkages to civil and commercial elements

- Theater communications that may be rapidly deployed, robust, and reliable, and that support all the military-unique requirements.

3.4.1 Communications Design Guidance

The following guidance is provided on communications:

- All communications requirements will be fulfilled using communications services and networks that adhere to the DISN architecture and the guidance provided in TAFIM Volume 2.

- Communications systems will provide compatibility with the Defense Message System (DMS).

- Where it is consistent with functional requirements, information systems will rely on DISN rather than providing their own communications capability.

3.4.2 Transition Elements

The current global communications network is a loosely connected collection of legacy and migration systems. In the near term, transition elements, such as application and network gateways, will provide greater connectivity and increased availability. In the mid to long term, transition elements will be phased out as all networks adopt accepted open systems elements. For the foreseeable future, the global communications network will remain a network of networks. The communications network will appear to be global. However, it will actually consist of a large number of smaller networks interconnected by gateways.

3.4.2.1 Multiple Protocol Networks

In a network of networks configuration, terminals, personal computers, and workstations will be directly connected to a local area network. The local area networks will be connected via a gateway to a metropolitan, regional, or wide area network. The metropolitan, regional, or wide area networks provide connectivity with other local area networks. In the near term, it is likely that many of these devices will use different communication protocols. Connectivity can be accomplished with multiple protocol networks. The multiple protocol network should not, however, be viewed as a long-term architectural solution. It provides connectivity directly among local networks with compatible protocol profiles but does not necessarily provide interoperability between local networks.

3.4.2.2 Techniques for Achieving Interoperability

Many existing systems are designed with proprietary protocols (e.g., IBM's prevalent System Network Architecture [SNA]). Bringing such systems into compliance with accepted open systems standards to achieve systemwide interoperability may be accomplished in several ways. Methods of achieving interoperability between different protocol systems include:

- **Total Adoption** – Changing all elements throughout the system at all architectural levels defined in Section 2 to operate with the specified standard protocol set (i.e., making the standard "native").
- **Conversion/Application Gateway** – Employing a gateway as an interface between two disparate network protocols to convert one to the other (e.g., IBM's SNA to TCP/IP). The gateway process involves performing all the functions of each protocol set (from the Network Switching through the Application Program levels) to retrieve the original application layer data and commands and then reintroducing it to the second protocol set. The application gateway is described further in Section 3.4.2.3.
- **Adaptation** – Substituting a different set of top layer protocols to ride on a lower layer standard (e.g., selecting a application set other than Telnet, File Transfer Protocol (FTP), or Simple Mail Transfer Protocol (SMTP) to work with TCP/IP). The process would incorporate several upper level protocol sets at a single processing location. It permits transmission with existing lower level protocols but with different Application layer protocols.
- **Multiple Stacks** – Equipping processors (workstations, servers, or mainframes) with several complete protocol sets. This gives the user the ability to enter all networks for which the processor retains a compatible protocol.
- **Encapsulation** – Wrapping the carrying protocol around the original protocol (e.g., TCP/IP wrapped around SNA) may be performed by a router. Encapsulation allows the carrying protocol to appear transparent (the original protocol enters and exits a network unaltered), permitting the original network elements to continue to operate without alteration.

3.4.2.3 Application Gateways

Application gateways may be used as a transition solution for the interconnection of two networks that adhere to different sets of standards. For example, application gateways could be used to connect a legacy network that supports military standard protocols to a network that is compliant with accepted open systems protocols. The gateway machine would be connected to both communications networks and support different connections on each end to enable the transfer of files. Applications that need to transfer files between the networks would use this application gateway and the gateway machine.

Application gateways are not beyond the current state of technology. There are platforms that have software installed to allow access by multiple protocols. For example, there are platforms that allow both FTAM and FTP access and others that support multiple protocols of electronic mail or messages.

Application gateways are not a solution and are not available for all application areas. If the protocols of the two applications are not sufficiently similar, the application gateway will not be viable. It will not be able to offer the functionality and robustness required. If either of the two applications does not have a significant market share, vendors will be hesitant to build an application gateway. That is, the gateway will not be commercially available.

3.4.2.4 Network Gateways

The global network will be made up of networks or communications links based on many different technologies. There will be links that are based on radio waves, optical beams, and fiber, coaxial, and copper cables. Communications is accomplished or optimized by using protocols that are uniquely matched to the network's or link's technology. The network gateway operates at the Network Switching level and is used to create the connection between links or networks based on different Network Switching level protocols. Thus, the network gateway is used to permit the compatible use of various technologies in the Transmission and Network Switching levels rather than facilitate Application level interoperability.

The network gateway performs packet translation. A number of protocols, which adhere to de jure and de facto standards, are based on a data packet. The network gateway understands how each of the protocols positions data within the packet. As it moves the packet from the source network to the destination network, it translates the data within the packet. Hence, each network only receives packets of the expected format. Network gateways that interconnect networks of different technologies and perform translation are viable and commercially available.

3.5 SECURITY PROTECTION GUIDANCE

3.5.1 Introduction

Volume 6 provides guidance to information system designers and implementors in the form of security principles and target security capabilities. The intent of this section is to provide a brief overview of selected guidance highlights of Volume 6 for individuals who are not information system security specialists. This section also contains two tables with general information about location of security services in the various OSI layer protocols and about appropriate mechanisms used to provide required security services. The source of these two tables is ISO 7498-2.

The guidance in Volume 6 is general because various mechanisms or combinations of mechanisms or services may be used or necessary to satisfy the requirements of the security policy for the information domain(s) handled in any particular information system. A wide variety of specific implementations, dictated by mission and threats, will be needed. The information system designer and implementor will need to work with the designated approving authority of the information system to identify the required level of protection and suitable mechanisms and services. In addition to mechanisms that may be implemented in the hardware and software of the information system, mechanisms that are doctrinal (i.e., physical, administrative, and personnel) will also be used to achieve the necessary level of security protection for the information domains handled by the information system.

The following factors should be considered in determining appropriate security mechanisms:

- Strength of security mechanisms
- Characteristics of security mechanisms
- Cost of security mechanisms
- Performance penalties.

As described in Section 2.5, implementation of security services and mechanisms may be allocated to the various components within a LSE and to the CN. The guidance focuses on implementation of security service mechanisms in end systems (or relay systems). Since end systems and relay systems are viewed as requiring the same kinds of security protection, guidance pertaining to an end system generally also applies to a relay system. The security protection provided in an end system will be implemented in both the hardware and software.

A minimum assurance analysis should be performed to satisfy the requirements of absolute protection as defined in Volume 6.

3.5.2 Guidance for End Systems and Relay Systems

A variety of choices exist for implementations of security mechanisms between the hardware and software portions of an end system or relay system.

3.5.2.1 Hardware Guidance

Implementations in hardware should:

- Enforce isolation of software functions by use of protected paths between users and applications and between application functions
- Ensure software and hardware integrity by use of anti-tampering and unwanted radiation devices/techniques
- Ensure availability by use of fault tolerant and fault detecting architectures.

Cryptographic mechanisms should be used for maintaining strict isolation for information in transfer between end systems. The cryptographic devices should be sufficiently flexible to support requirements of different information domains. It may be necessary to use multiple cryptographic devices.

3.5.2.2 Software Guidance

A software architecture built around trusted and untrusted software components is recommended. Trusted software should be used for security-critical functions, including exchanges between information domains. Trusted software must be evaluated and maintained under strict configuration management. Untrusted software should only be able to invoke functions through the use of trusted software. Nonetheless, it is recommended that untrusted software be obtained from reliable sources, tested before use, and be subjected to integrity safeguards to preclude its modification. Configuration management should also be applied to it. Security protection is provided for guidance in three service areas of the end system software:

- Operating system services
- Network services
- System management services.

3.5.2.2.1 Operating System Services

The recommended end system security architecture relies upon an engineering approach that seeks to isolate security-critical functions into relatively small modules that are related in well-defined ways. Security-critical functions should generally provide commonly used, low-level operating system functions. This is considered consistent with commercial operation system vendors' design and implementation strategies. Volume 6 identifies some security-critical functions. Prototyping and experimentation is also needed to identify other software functions that need to be handled as security-critical.

3.5.2.2.2 Network Services

Communications protocols are to be used for implementing security protection mechanisms for inter-end-system information transfers within the same information domain. The allocation of security services to the various OSI layers is shown in Figure 3-1. As shown, no security services are allocated to OSI layer 5, and no specific services are allocated to layer 6. It also needs to be noted that all services are allocated for possible implementation in OSI layer 7, the application layer. However, the implementation may not be in OSI layer 7, but rather in the application process using the communications services. Details of the rationale for allocation of the security services to the OSI layers are contained in ISO 7498-2. Security protocols relevant to the layers shown in Figure 3-1 are given in Volumes 6 and 7.

Some lower layer security protocols can multiplex several security associations between the same end systems. It is not expected, however, that multiplexing for information systems handling different information domains simultaneously will be acceptable to a designated approving authority.

Service	Layer 1	Layer 2	Layer 3	Layer 4	Layer 7**
Authentication: Peer Entity and Data Origin	N	N	Y	Y	Y
Access Control	N	N	Y	Y	Y
Confidentiality: Connection Oriented	Y	Y	Y	Y	Y
Confidentiality: Connectionless	N	Y	Y	Y	Y
Confidentiality: Selective Field	N	N	N	N	Y
Traffic Flow Confidentiality	Y	N	Y	N	Y
Integrity: Connection Oriented With Recovery	N	N	N	Y	Y
Integrity: Connection Oriented Without Recovery	N	N	Y	Y	Y
Integrity: Selective Field and Connection Oriented	N	N	N	N	Y
Integrity: Connectionless	N	N	Y	Y	Y
Integrity: Selective Field Connectionless	N	N	N	N	Y
Non-repudiation: Origin or Delivery	N	N	N	N	Y

Key

* No services are allocated to OSI layer 5; layer 6, the presentation layer, contains a number of security facilities which support the provision of security services by the application layer.

** The services allocated to OSI layer 7, the application layer, may be provided by the application process itself.

N = Service not allocated to layer.

Y = Service allocated and should be provided for in layer protocol.

NOTE: This table needs to be revised to reflect pending changes to ISO 7498-2. This revision will be accomplished when the source information is finalized and analyzed.

Figure 3-1. Security Services Allocated to OSI Layers*

3.5.2.2.3 System Management Services

A security management application process should be used for establishing a security association for interactive communications among end systems. Implementation of communications applications (e.g., X.400 electronic mail, X.500 directory services, file transfer) and communications protocols will occur as untrusted applications within the end system software security architecture. Security protection for these untrusted applications should be provided by the establishment of a security association for an interactive communications dialog. A SMAP should be used to establish a security association.

End systems that support multiple information domains must also provide independent security management for each of the information domains. The security policy rules for both end system security management and information domain security management must be part of the end system security management information base. The relationship between the SMIB and a SMAP is described in Volume 6. The SMAP must be capable of responding to an end user application request for a specific security mechanism or be able to adopt a suitable one based on the information domain or end system security policies contained in the SMIB.

To allow for effective distribution of security management across many end system platforms, standardization of security management functions, data structures, and protocols is recommended. Specific areas for security management standardization are identified in Volume 6.

3.5.3 Guidance for Architectural Components Other Than End Systems or Relay Systems

This section provides a brief overview of the guidance in Volume 6 for the architectural components of local subscriber environment, local communications system, and communications network.

3.5.3.1 Local Communications System Guidance

Security services are generally not required of implementations in the LCS unless the LCS is only used for communications among end systems in the same LSE. Even if this condition is satisfied, care must be taken that implementations in the LCS do not interfere with requirement added at a later date for communications with end systems in other LSEs. Nonetheless, should implementations of security mechanisms in the LCS be desirable, use of the same approaches (protocols and security management applications) as described for the end system network services will apply.

3.5.3.2 Communications Network Guidance

Because of the use of common carriers for transmitting information, the CNs are expected frequently not to be under the control of the DoD, and perhaps not under the control of a DoD organization with a comparable or otherwise suitable DoD information domain security policy. Therefore, allocating security services other than availability to the CNs is not recommended. In addition to the general need for communications resource availability, this may also provide for protection against "denial of service" to specific applications.

APPENDIX A

REFERENCES

Note: References appearing in this section represent documents used in preparation of the TAFIM, including some sources used at the time of initial document development that may no longer be current or applicable. The reader is advised to check the current applicability of a reference appearing in this list before using it as an information source. The reference section will be completely reviewed and revised for the next release of the TAFIM.

1. Air Force Communications and Computer Systems Integration Guide, U.S. Air Force Technology Integration Center, Version 3, April 1991.

2. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. VI, Integrated Systems Control April 1987.

3. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. VII, Software Architecture, December 1990.

4. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. I, Overview, July 1990.

5. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. IV, Local Information Transfer, September 1987.

6. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. V, Long Haul Information Transfer, April 1987.

7. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. II, Deployable Architecture, September 1989.

8. Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. III, Data Management Architecture, April 1990.

9. Air Force Planning & Architecture Guidance, Part II Architecture & Implementation Guide, U.S. Air Force, March 1990.

10. American National Standard for Information Systems - Dictionary for Information Systems, ANSI X3.172-1990, American National Standards Institute, July 1990.

11. Application Portability Profile, National Institute of Standards and Technology, April 1991, Army Tactical Command and Control Information System (ATCCIS), Working Paper 30, Applicability of the Architecture, January 1990.

12. Army Tactical Command and Control Information System (ATCCIS), Working Paper 22, Architectural Concepts, September 1987.

13. Army Tactical Command and Control Information System (ATCCIS), Working Paper 24, Architectural Definition, September 1990.

14. Army Tactical Command and Control Information System (ATCCIS), Working Paper 34, ATCCIS Communications, January 1990.

15. Army Tactical Command and Control Information System (ATCCIS), Working Paper 11, Functional Requirements Derived From Key Tasks, January 1990.

16. Army Tactical Command and Control Information System (ATCCIS), Working Paper 7N, Standardization of Data for Interoperability, September 1990.

17. Army Tactical Command and Control Information System (ATCCIS), Working Paper 25, Technical Standards for Command and Control Information Systems (CCISs), Edition 3, January 1992.

18. Army Command and Control Information Systems Commonalty with the Information Systems Architecture, Draft, U.S. Army Information Systems Engineering Command, April 1992.

19. Army Information Architecture, Department of the Army, Pamphlet 25-1, 20 August 1991.

20. Army Information Mission Area Information System Architecture Security, Draft, U.S. Army Information Systems Engineering Command, September 1991.

21. Army Information Systems Architecture Circa 1997, U.S. Army Information Systems Engineering Command, August 1989.

22. Army Information Systems Architecture (ISA), briefing, June 1992. Army Information Systems Architecture '97, Strategic Implementation Plan, U.S. Army Information Systems Engineering Command, April 1992.

23. Army Information Systems Architecture, Strawman Version, U.S. Army ISC, 30 March 1992.

24. Army Information Systems Architecture, Vol. II, Strategic and Sustaining Base Architecture, U.S. Army ISC, December 1991.

25. Army Information Systems Architecture, Vol. III, Technology and Standards, U.S. Army ISC, December 1991.

26. Army, Integrated Architecture Volume Mid-Range Technical Architecture, U.S. Army Information Systems Engineering Command, July 1991.

27. Army, Multimedia for the Information Systems Architecture, Coordination Draft, U.S. Army Information Systems Engineering Command, April 1992.

28. Atkinson, Malcolm et al., December 1989, "The Object-Oriented Database System Manifesto," *Proc. DOOD 1989.*

29. Berson, Alex, 1992, *Client/Server Architecture*, New York: McGraw-Hill, Inc.

30. Boeing Enterprise Network, Vol. I, Vision and Architecture, Boeing Co., November 1989.

31. Boeing Enterprise Network, Vol. II, General Guidelines and Principles for Transition to BNA Phase 2, Boeing Co., November 1989.

32. CALS Architecture Study, Vol. II, The Joint CALS Management Office, 30 June 1991.

33. CALS Architecture Study, Vol. I: Report, The Joint CALS Management Office, 30 June 1991.

34. CCIS, Command and Control Information System (CCIS) Architecture, MITRE, February 1990. CCIS, Generic and Target Architecture for Command and Control Information Systems, IDA Paper P-2490, September 1991.

35. CCIS, Survey of Technical Standards for Command and Control Information Systems, IDA Paper P-2457, September 1991.

36. CIM Human Computer Interface Style Guide, Version 1.0, February 1992.

37. CIM Review of Software Architectures, MITRE Corp., February 1992.

38. CIM Software Architecture Framework (Draft), MITRE Corp., April 1992.

39. CIM Software Development Framework, Draft, Center for Information Management, May 1992.

40. CIM Technical Reference Model for Information Management, Version 1.2, Center for Information Management, May 1992.

41. Copernicus Architecture, Implementation Plan for Phase II, Space and Naval Warfare Systems Command, December 1991.

42. Copernicus Architecture, Phase I: Requirements Definition, U.S. Navy, August 1991.

43. Copernicus Architecture, Phase I: Requirements Definition, Space and Naval Warfare Systems Command, December 1991.

44. Counter Narcotics, Information Protection Architecture, draft, Office of National Drug Control Policy, November 1991.

45. C4I For The Warrior Interoperability Tiger Team Final Report, Joint Staff, May 1992.

46. Defense Information Systems Agency, Client Server Migration Guidance for the Mission Support Area, Version 2.0, September 1993.

47. Defense Information Systems Agency, Defense Information System Network (DISN), A Goal Integrated Communications Architecture and Transition Strategy, Interim Report, April 1992.

48. Defense Information Systems Agency, Defense Information System Network (DISN), Final Report, September 1990.

49. Defense Logistics Agency (DLA), DoD Open Systems Life Cycle Management Concept for Corporate Information Management, October 1991.

50. Defense Logistics Agency (DLA), Office of Information Systems and Technology, Open Network Systems Implementation and Management, Version 1.1, September 1991.

51. Defense Logistics Agency (DLA), Office of Information Systems and Technology, Information Resources Management Environment Vision and Prescription, Version 1.1, April 1991.

52. Defense Logistics Agency (DLA), Systems Software Blueprint, DSAC System Software, June 1986.

53. Defense Logistics Agency (DLA), The DLA Open Systems Architecture for Information Systems, DSAC, December 1988.

54. Department of Defense (DoD) Command, Control, Communications, Computers, and Intelligence (C4I) for the Warrior Directive, draft, April 1992.

55. Department of Defense (DoD) Goal Security Architecture (DGSA) Executive Summary, draft, August 1993.

56. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) A Framework for Evolution of the Department of Defense Intelligence Information System (DoDIIS), Defense Intelligence Agency, July 1991.

57. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Client-Server Environment (CSE) Specification, The Engineering Review Board of the DoDIIS Management Board of the Defense Intelligence Agency, June 1991.

58. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Reference Model for the 1990s, DoDIIS Management Board, October 1991.

59. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Standards Document, The MITRE Corporation, October 1991.

60. Department of Defense (DoD) Intelligence Information Systems (DoDIIS) Style Guide, DoDIIS Management Board, October 1991.

61. Department of Defense (DoD) Software Technology Strategy, draft, Director of Defense Research and Engineering, December 1991.

62. Department of Defense Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, Department of Defense, December 1985.

63. DSAC Office of Architectural Integration, Strategic Architectural Objective, Systems Capacity Management, June 1989.

64. DSAC Office of Architectural Integration, Strategic Architectural Objective, DLA Communications, June 1989.

65. DSAC Office of Architectural Integration, Strategic Architectural Objective, Corporate Data System Application Design Objectives for Departmental and Personal Platforms, August 1990.

66. DSAC Office of Architectural Integration, Strategic Architectural Objective, Data Center Automation, June 1989.

67. DSAC Office of Architectural Integration, Strategic Architectural Objective, Operating Systems, January 1990.

68. DSAC Office of Architectural Integration, Strategic Architectural Objective, DLA Communications Configuration, June 1990.

69. DSAC Office of Architectural Integration, Strategic Architectural Objective, Systems Information Management, June 1989.

70. DSAC Office of Architectural Integration, Strategic Architectural Objective, Development Support, June 1989.

71. DSAC Office of Architectural Integration, Strategic Architectural Objective, Database Management System July 1990.

72. DSAC Office of Architectural Integration, Strategic Architectural Objective, Data Administration, January 1990.

73. Faulkner, March 1993, "Data Base Management Systems," *Faulkner Technical Report.*

74. FIPS Publication 11-3, Guideline: American National Dictionary for Information System, National Institute of Standards and Technology, February 1991.

75. FIPS Publication 146-2, Profiles for Open Systems Internetworking Technologies (POSIT), National Institute of Standards and Technology, 1996.

76. Functional Process Improvement, DoD 8020.1-M, draft, April 1992.

77. Global Transportation Network, C4S Technical Standards, U.S. Transportation Command, 15 April 1992.

78. Information Technology Portfolio Matrix, draft, June 1992.

79. International Organization for Standardization (ISO), 1984, Information Processing Systems, *Open Systems Interconnection Reference Model: Basic Reference Model*, ISO 7498.

80. International Organization for Standardization (ISO), 1989, Information Processing Systems, *Open Systems Interconnection Reference Model, Part 4: Management Framework*, ISO 7498.

81. JCS, Command Center System Architecture and TA/CE Guidance, FY 92-97, Volume I, JCS, September 1991

82. JCS, Joint Staff Automation for the Nineties (JSAN), Grumman Data Systems Corporation, March 1991.

83. MAGTF, Interoperability Requirements Concepts (MIRC), USMC, 4 May 1990.

84. Marine Corps Tactical Communications Architecture (MCTCA), DON, HQMC, 30 July 1990.

85. Nation Photographic Interpretation Center (NPIC) Information System Volume I, Architecture Definition, NPIC, December 1991.

86. Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture, International Organization for Standardization.

87. POSIX, Draft Guide to the POSIX Open Systems Environment, P1003.0/D15, draft, June 1992.

88. Software Technology for Adaptable, Reliable Systems (STARS), Updated System Specification (SSS), September 1990.

89. Standards-Based Architecture Planning Guide, Draft Version 1.2, DMR Group, Inc., 24 April 1992.

90. Strategies for Open Systems, Stage Four, Standards-Based Architectures, DMR Group, 1991.

91. Target Architecture and Implementation Strategy for the Joint MLS Technology Insertion Program, MTR-91W00134, The MITRE Corporation, September 1991.

92. UNIX, 1992 Road Map for System V and Related Technologies, UNIX International, February 1992.

93. UNIX, 1992 UNIX System V Release 4 Product Catalog, UNIX International, Spring 1992

APPENDIX B

ACRONYMS AND DEFINITIONS

ACRONYMS

AIS	Automated Information System
ANSI	American National Standards Institute
API	Application Program Interface
APP	Application Portability Profile
BBS	Bulletin Board System
C3I	Command, Control, Communications, and Intelligence
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CASE	Computer-Aided Software Engineering (See ISEE)
CIM	Corporate Information Management
CJCS	Chairman of the Joint Chiefs of Staff
CN	Communications Network
CORBA	Common Object Request Broker Architecture
COTS	Commercial-Off-the-Shelf
CODASYL	Conference on Data Systems Languages
DBMS	Database Management System
DD/DS	Data Dictionary/Directory System
DGSA	Defense Goal Security Architecture
DISA	Defense Information Systems Agency
DMS	Defense Message System
DoD	Department of Defense
DoDD	Department of Defense Directive
DISN	Defense Information Systems Network
E-mail	Electronic Mail
EEI	External Environment Interface
ES	End System
FIPS	Federal Information Processing Standard
FTAM	File Transfer, Access, and Management

FTP	File Transfer Protocol
GOSIP	Government Open System Interconnection Profile
GSS	General Security Service
GUI	Graphical User Interface
IBM	International Business Machines
IETF	Internet Engineering Task Force
IM	Information Management
IRDS	Information Resource Dictionary System
ISAM	Indexed Sequential Access Method
ISO	International Organization for Standardization
IT	Information Technology
ITSI BBS	Information Technology Standards Information Bulletin Board System
JIEO	Joint Interoperability and Engineering Organization
LAN	Local Area Network
LCS	Local Communications System
LSE	Local Subscriber Environment
MAN	Metropolitan Area Network
MHS	Message Handling System
MIL-STD	Military Standard
MOP	Memorandum of Policy
NCSC	National Computer Security Center
NIST	National Institute of Standards and Technology
OLE	Object Linking and Embedding
OODBMS	Object-Oriented Database Management System
ORB	Object Request Broker
OSI	Open System Interconnection
OMG	Object Management Group
POSIT	Profiles for Open Systems Internetworking Technologies
POSIX	Portable Operating System Interface (for Computer Environments)
RDBMS	Relational Database Management System
RS	Relay System

SAMP	Security Association Management Protocol
SMAP	Security Management Application Process
SMIB	Security Management Information Base
SMTP	Simple Mail Transfer Protocol
SNA	System Network Architecture
SQL	Structured Query Language
SWG	Special Working Group
TAFIM	Technical Architecture Framework for Information Management
TCP/IP	Transmission Control Protocol/Internet Protocol
TRM	Technical Reference Model
WAN	Wide Area Network
WWW	World Wide Web

DEFINITIONS

Application–The use of capabilities (services and facilities) provided by an information system specific to the satisfaction of a set of user requirements. [P1003.0/D15]

Application Platform–The collection of hardware and software components that provide the services used by support and mission-specific software applications.

Application Portability Profile (APP)–The structure that integrates Federal, national, international, and other specifications to provide the functionality necessary to accommodate the broad range of federal information technology requirements. [APP]

Application Program Interface (API)–(1) The interface, or set of functions, between the application software and the application platform. [APP] (2) The means by which an application designer enters and retrieves information.

Architecture–Architecture has various meanings depending upon its contextual usage. (1) The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time. [IEEE STD 610.12] (2) Organizational structure of a system or component. [IEEE STD 610.12]

Architecture, Database–The logical view of the data models, data standards, and data structure. It includes a definition of the physical databases for the information system, their performance requirements, and their geographical distribution. Ref DoD 8020.1-M, Appendix J

Architecture Target–Depicts the configuration of the target open information system. Ref DoD 8020.1-M

Architecture, Infrastructure–Identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities to include hardware builds versus schedule and costs. [DoD 8020.1-M, Appendix J specifically paragraph 5(14)(c), Table J-2]

Architectural Structure–Provides the conceptual foundation of the basic architectural design concepts, the layers of the technical architecture, the services provided at each layer, the relationships between the layers, and the rules for how the layers are interconnected.

Automated Information System (AIS)–Computer hardware, computer software, telecommunications, information technology, personnel, and other resources that collect, record, process, store, communicate, retrieve, and display information. An AIS can include computer software only, computer hardware only, or a combination of the above. [DoDD 8000.1]

Baseline–A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development and that can be changed only through formal change control procedures or a type of procedure such as configuration management. [IEEE STD 610.12]

Commercial-Off-The-Shelf (COTS)–Refers to an item of hardware or software that has been produced by a contractor and is available for general purchase. Such items are at the unit level or higher. Such items must have been sold and delivered to government or commercial customers, must have passed customer's acceptance testing, be operating under customer's control, and within the user environment. Further, such items must have meaningful reliability, maintainability, and logistics historical data.

Communications Link–The cables, wires, or paths that the electrical, optical, or radio wave signals traverse. [TA]

Communications Network–A set of products, concepts, and services, that enable the connection of computer systems for the purpose of transmitting data and other forms (e.g., voice and video) between the systems.

Communications Node–A node that is either internal to the communications network (e.g., routers, bridges, or repeaters) or located between the end device and the communications network to operate as a gateway. [TA]

Communications Services–A service of the Support Application entity of the Technical Reference Model that provides the capability to compose, edit, send, receive, forward, and manage electronic and voice messages and real time information exchange services in support of interpersonal conferencing. [TA]

Communications System–A set of assets (transmission media, switching nodes, interfaces, and control devices) that will establish linkage between users and devices.

Configuration Management–A discipline applying technical and administrative direction and surveillance to: (a) identify and document the functional and physical characteristics of a configuration item, (b) control changes to those characteristics, and (c) record and report changes to processing and implementation status.

Connectivity Service–A service area of the External Environment entity of the Technical Reference Model that provides end-to-end connectivity for communications through three transport levels (global, regional, and local). It provides general and applications-specific services to platform end devices. [TA]

Database Utility Service–A Service of the Support Application Entity of the Technical Reference Model that provides the capability to retrieve, organize, and manipulate data extracted from a database. [TA]

Data Dictionary–A specialized type of database containing metadata, which is managed by a data dictionary system; a repository of information describing the characteristics of data used to design, monitor, document, protect, and control data in information systems and databases; an application of data dictionary systems. [DoDD 8320.1]

Data Element–A basic unit of information having a meaning and that may have subcategories (data items) of distinct units and values. [DoDD 8320.1]

Data Interchange Service–A service of the Platform entity of the Technical Reference Model that provides specialized support for the interchange of data between applications on the same or different platforms. [TA]

Data Management Service–A service of the Platform entity of the Technical Reference Model that provides support for the management, storage, access, and manipulation of data in a database. [TA]

Directory Service–A service of the External Environment entity of the Technical Reference Model that provides locator services that are restricted to finding the location of a service, location of data, or translation of a common name into a network specific address. It is analogous to telephone books and supports distributed directory implementations. [TA]

Distributed Database–(1) A database that is not stored in a central location but is dispersed over a network of interconnected computers. (2) A database under the overall control of a central database management system but whose storage devices are not all attached to the same processor. (3) A database that is physically located in two or more distinct locations. [FIPS PUB 11-3]

Enterprise–The highest level in an organization – includes all missions and functions. [TA]

Enterprise Model–A high-level model of an organization's mission, function, and information architecture. The model consists of a function model and a data model.

External Environment Interface (EEI)–The interface that supports information transfer between the application platform and the external environment. [APP]

Functional Architecture–The framework for developing applications and defining their interrelationships in support of an organization's information architecture. It identifies the major functions or processes an organization performs and their operational interrelationships. [DoD 5000.11-M]

Functional Area–A range of subject matter grouped under a single heading because of its similarity in use or genesis. [DoDD 8320.1]

Function–Appropriate or assigned duties, responsibilities, missions, tasks, powers, or duties of an individual, office, or organization. A functional area is generally the responsibility of a PSA (e.g., personnel) and can be composed of one or more functional activities (e.g., recruiting), each of which consists of one or more functional processes (e.g., interviews). Ref Joint Pub 1-02, DoDD 8000.1, and DoD 8020-1M.

Functional Activity Program Manager (FAPM)–FAPMs are designated by PSAs and are accountable for executing the functional management process. Supported by functional representatives from the DoD Components, FAPMs develop functional architectures and strategic plans, and establish the process, data, and information system baselines to support functional activities within the functional area Ref DoD 8020.1-M Ch 1 B(2).

Functional Data Administrator (FDAd)–OSD PSAs exercise or, designate functional data administrators to perform data administrator responsibilities to support execution of the functional management process, and to function within the scope of their overall assigned responsibilities. Ref DoDD 8320.1 and DoD 8020.1-M, Appendix A.

Functional Economic Analysis (FEA)–A structured proposal that serves as the principal part of a decision package for enterprise (individual, office, organization - see function) leadership. It includes an analysis of functional process needs or problems; proposed solutions, assumptions, and constraints; alternatives; life-cycle costs; benefits and/or cost analysis; and investment risk analysis. It is consistent with, and amplifies, existing DoD economic analysis policy. Ref DoDI 7041.3, DoDD 8000.1, and DoD 8020.1-M, Appendix H.

Hardware–(1) Physical equipment, as opposed to programs, procedures, rules, and associated documentation. (2) Contrast with software. [FIPS PUB 11-3]

Information–Any communication or representation of knowledge such as facts, data, or opinions, in any medium or form, including textual, numerical, graphic, cartographic, narrative, or audiovisual forms. [OMB CIRC A-130]

Information Domain–A set of commonly and unambiguously labeled information objects with a common security policy that defines the protections to be afforded the objects by authorized users and information management systems. [DISSP]

Information Management(IM)–The creation, use, sharing, and disposition of information as a resource critical to the effective and efficient operation of functional activities. The structuring of functional processes to produce and control the use of data and information within functional activities, information systems, and computing and communications infrastructures. [DoDD 8000.1]

Information Resources Management (IRM)–The planning, budgeting, organizing, directing, training, promoting, controlling, and management activities associated with the burden (cost), collection, creation, use, and dissemination of information by Agencies and includes the management of information and related resources, such as federal information processing (FIP) resources. Ref PL No 99-591, DoDD 8000.1.

Information Technology (IT)–The technology included in hardware and software used for Government information, regardless of the technology involved, whether computers, communications, micro graphics, or others. Ref OMB Circular A-130 and DoDD 8000.1.

Infrastructure–Infrastructure is used with different contextual meanings. Infrastructure most generally relates to and has a hardware orientation but note that it is frequently more comprehensive and includes software and communications. Collectively, the structure must meet the performance requirements of and capacity for data and application requirements. Again note that just citing standards for designing an architecture or infrastructure does not include functional and mission area requirements for performance. Performance requirement metrics must be an inherent part of an overall infrastructure to provide performance interoperability and compatibility. It identifies the top-level design of communications, processing, and operating system software. It describes the performance characteristics needed to meet database and application requirements. It provides a geographic distribution of components to locations. The infrastructure architecture is defined by the service provider for these capabilities. It includes processors, operating systems, service software, and standards profiles that include network diagrams showing communication links with bandwidth, processor locations, and capacities to include hardware builds versus schedule and costs. Ref DoD 8020.1-M

Interoperability–The ability of systems to exchange useful data and information.

Legacy Environments–Legacy environments could be called legacy architectures or infrastructures and as a minimum consist of a hardware platform and an operating system. Legacy environments are identified for phase-out, upgrade, or replacement. All data and applications software that operate in a legacy environment must be categorized for phase-out, upgrade, or replacement.

Legacy Systems–Systems that are candidates for phase-out, upgrade, or replacement. Generally legacy systems are in this category because they do not comply with data standards or other standards. Legacy system workloads must be converted, transitioned, or phased out (eliminated). Such systems may or may not operate in a legacy environment.

Life Cycle–The period of time that begins when a system is conceived and ends when the system is no longer available for use. [IEEE STD 610.12]. AIS life cycle is defined within the context of life-cycle management in various DoD publications. It generally refers to the usable system life.

Local Area Network (LAN)–A data network, located on a user's premises, within a limited geographic region. Communication within a local area network is not subject to external regulation; however, communication across the network boundary may be subject to some form of regulation. [FIPS PUB 11-3].

Migration Systems–An existing AIS, or a planned and approved AIS, that has been officially designated to support common processes for a functional activity applicable to use DoD-wide or DoD Component-wide. Systems in this category, even though fully deployed and operational, have been determined to accommodate a continuing and foreseeable future requirement and, consequently, have been identified for transitioning to a new environment or infrastructure. A migration system may need to undergo transition to the standard technical environment and standard data definitions being established through the Defense IM Program, and must "migrate" toward that standard. In that process it must become compliant with the Reference Model and the Standards Profile. A system in this category may require detailed analysis that involves a total redesign, reprogramming, testing, and implementation because of a new environment and how the "users" have changed their work methods and processes. The detailed analysis may identify the difference between the "as is" and the "to be" system. [DoD 8020.1-M].

Multimedia Service–A service of the TRM that provides the capability to manipulate and manage information products consisting of text, graphics, images, video, and audio. [TA]

Open Specifications–Public specifications that are maintained by an open, public consensus process to accommodate new technologies over time and that are consistent with international standards. [P1003.0/D15]

Open System–A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered applications software: (a) to be ported with minimal changes across a wide range of systems, (b) to interoperate with other applications on local and remote systems, and (c) to interact with users in a style that facilitates user portability. [P1003.0/D15]

Open Systems Environment (OSE)–The comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability or for portability of applications, data, or people, as specified by information technology standards and profiles. [P1003.0/D15]

Operating System Service–A core service of the Platform entity of the Technical Reference Model that is needed to operate and administer the application platform and provide an interface between the application software and the platform (e.g., file management, input/output, print spoolers). [TA]

Platform–The entity of the Technical Reference Model that provides common processing and communication services that are provided by a combination of hardware and software and are required by users, mission area applications, and support applications. [TA]

Portability–(1) The ease with which a system or component can be transferred from one hardware or software environment to another. [IEEE STD 610.12] (2) A quality metric that can be used to measure the relative effort to transport the software for use in another environment or to convert software for use in another operating environment, hardware configuration, or software system environment. [IEEE TUTOR] (3) The ease with which a system, component, data, or user can be transferred from one hardware or software environment to another. [TA]

Process Model–Provides a framework for identifying, defining, and organizing the functional strategies, functional rules, and processes needed to manage and support the way an organization does or wants to do business--provides a graphical and textual framework for organizing the data and processes into manageable groups to facilitate their shared use and control throughout the organization. [DoD 5000.11-M]

Profile–A set of one or more base standards, and, where applicable, the identification of those classes, subsets, options, and parameters of those base standards, necessary for accomplishing a particular function. [P1003.0/D15]

Profiling–Selecting standards for a particular application. [P1003.0/D15]

Scalability–The ability to use the same application software on many different classes of hardware/software platforms from personal computers to super computers (extends the portability concept). [USAICII]. The capability to grow to accommodate increased work loads.

Seamless Interface–Ability of facilities to call one another or exchange data with one another in a direct manner. Integration of the user interface that allows a user to access one facility through another without any noticeable change in user interface conventions. [DSAC SYS IM]

Stovepipe System–A system, often dedicated or proprietary, that operates independently of other systems. The stovepipe system often has unique, nonstandard characteristics.

System–People, machines, and methods organized to accomplish a set of specific functions. [FIPS PUB 11-3]

System Management Service–A service of the Platform entity of the TRM that provides for the administration of the overall information system. These services include the management of information, processors, networks, configurations, accounting, and performance. [TA]

Technical Reference Model (TRM)–The document that identifies a target framework and profile of standards for the DoD computing and communications infrastructure. [TRM]

User–(1) Any person, organization, or functional unit that uses the services of an information processing system. (2) In a conceptual schema language, any person or any thing that may issue or receive commands and messages to or from the information system. [FIPS PUB 11-3]

User Interface Service–A service of the Platform entity of the Technical Reference Model that supports direct human-machine interaction by controlling the environment in which users interact with applications. [TA]

REFERENCES FOR DEFINITIONS

[AF 700-50,V] Air Force Communications-Computer Systems Architecture, AF Pamphlet 700-50, Vol. V, Long Haul Information Transfer, April 1987.

[APP] NIST Special Report, APP: The US. Governments Open System Environment Profile, Draft, January 1991.

[ARMY 25-1] Army Information Architecture, Department of Army, Pamphlet 25-1, 20 August 1991.

[BOEING] Boeing Enterprise Network, Vol. I, Vision and Architecture, Boeing Co., November 1989.

[DATE] Date, C. J. with Colin J. White, A Guide to DB2, Third Edition, Addison-Wesley Publishing Co., Reading, MA, 1989.

[CIM] DRAFT CIM Architecture Framework, November 1991.

[DISSP] Defense Information Systems Security Program

[DoD 5000.11-M] Department of Defense, DoD Data Administration Procedures, Department of Defense Manual 5000. 11-M Draft, 30 June 1991.

[DoD 8320.1-M] Department of Defense Data Administration Working Group, DoD Data Administration Procedures Manual, DoD Manual 8320. 1-M Draft, 25 October 1991.

[DoDD 5000.29] Department of Defense, Management of Computer Resources in Major Defense Systems, Department of Defense Directive 5000.29, 26 April 1976.

[DoDD 8000.1] Defense Information Management (IM) Program, Department of Defense Directive 8000.1, 27 October 1992.

[DoDD 8320.1] Department of Defense Data Administration, Department of Defense Directive 8320.1, 26 September 1991.

[DSAC SYS IM] DSAC Office of Architectural Integration, Strategic Architectural Objective, Systems Information Management, June 1989.

[ELMAGARMID] Elmagarmid, Ahmed K., and Calton Pu, "Guest Editors' Introduction to the Special Issue of Heterogeneous Databases," ACM Computing Surveys, Volume 22, Number 3, September 1990.

[FIPS PUB 11-3] FIPS Publication 11-3, Guideline American National Dictionary for Information System, National Institute of Standards and Technology, February 1991.

[FRAMEWORK] Software Development Framework, Draft, 15 May 1992.

[HCI] DISA, Human Computer Interface Style Guide, Version 1.0, 12 February 1992.

[IEEE STD 610.12] Institute of Electrical and Electronics Engineers, Inc., IEEE Standard Glossary of Software Engineering Terminology, IEEE STD 610.12-1990, 10 December 1990.

[IEEE TUTOR] Standards, Guidelines, and Examples on System and Software Requirements Engineering, Merline Dorman and Richard Thayer, editors, IEEE Computer Society Press Tutorial, 1990.

[KORTH] Korth, Henry F. and Abraham Silberschatz, Database System Concepts, McGraw-Hill Book Company, New York, 1986.

[MCC] McClure, Carma, The Three R's of Software Automation," Prentice Hall, 1992.

[OMB CIRC A-130] Office of Management and Budget Circular A-130, "Management of Information Resources," 1985.

[P1003.0/D15] Technical Committee on Operating Systems and Application Environments of the IEEE Computer Society, "Standards Project Draft Guide to the POSIX Open Systems Environment," June 1992.

[STALLINGS] Stallings, William, Business Data Communications, MacMillan Publishing Company, New York, 1990.

[TA] DoD TAFIM for Information Management (Draft).

[TRANSCOM] Global Transportation Network, C4S Technical Standards, US. Transportation Command, 15 April 1992.

[TRM] CIM Technical Reference Model for Information Management, Version 1.2, Center for Information Management, May 1992

[USAICII] Army Information Systems Architecture, Vol. II, Strategic and Sustaining Base Architecture, US. Army ISC, December 1991.

[WEBSTER] Webster's II New Riverside University Dictionary, The Riverside Publishing Company, MA, 1988.

APPENDIX C

OPEN SYSTEMS

A critical objective of the DoD Information Management initiative is the implementation of a computing and communications infrastructure that supports portability, interoperability, and scalability. To achieve this objective the DoD must develop and use open systems. The following definitions apply to open systems:

- OPEN SYSTEM: A system that implements sufficient open specifications for interfaces, services, and supporting formats to enable properly engineered hardware and applications software to:
  - Interoperate with other applications on local and remote systems
  - Be ported with minimal changes across a wide range of systems
  - Interact with users in a style that facilitates user portability
  - Enable users to increase processing power as their functional needs grow, without the need to re-write applications (i.e., scalability)
- OPEN SPECIFICATION: Public specifications that are maintained by an open, public consensus process to accommodate new technologies over time and that are consistent with international standards.
- OPEN SYSTEM ENVIRONMENT (OSE): The comprehensive set of interfaces, services, and supporting formats, plus user aspects for interoperability and scalability, or for portability of applications, data, or people, as specified by information technology standards and profiles.
- OPEN SYSTEM ARCHITECTURE: The framework describing the entities (e.g., components, services) and their interrelationships in an open system.

Open systems environments and architectures are intended to help achieve portability, interoperability, scalability and cost effectiveness of systems. These attributes facilitate technology insertion and rapid system evolution to respond to changing functional practices – functional and technical managers will have the capability to selectively preserve or reconfigure parts of the infrastructure based on functional needs.

Open systems are modular, enabling users to define, acquire, and add to systems that are supplied by a variety of vendors in an open, competitive market. An open system supports the interoperability of hardware, software, and communications products developed by different suppliers at different times.

DoD information systems will incrementally evolve to converge towards open system architecture guidelines and standards while accommodating existing baselines and transition environments. Implementation guidelines will be provided for engineering and economic analysis of options and opportunities to evolve baselines to target architectures. They will support the decision making process to select the best overall targets and transition paths.

APPENDIX D

PROPOSING CHANGES TO TAFIM VOLUMES

D.1 INTRODUCTION

Changes to the TAFIM will occur through changes to the TAFIM documents (i.e., the TAFIM numbered volumes, the CMP, and the PMP). This appendix provides guidance for submission of proposed TAFIM changes. These proposals should be described as specific wording for line-in/line-out changes to a specific part of a TAFIM document.

Use of a standard format for submitting a change proposal will expedite the processing of changes. The format for submitting change proposals is shown in Section D.2. Guidance on the use of the format is provided in Section D.3.

A Configuration Management contractor is managing the receipt and processing of TAFIM change proposals. The preferred method of proposal receipt is via e-mail in ASCII format, sent via the Internet. If not e-mailed, the proposed change, also in the format shown in Section D.2, and on both paper and floppy disk, should be mailed. As a final option, change proposals may be sent via fax; however, delivery methods that enable electronic capture of change proposals are preferred. Address information for the Configuration Management contractor is shown below.

Internet: **tafim@bah.com**

Mail: **TAFIM**
Booz Allen & Hamilton Inc.
5201 Leesburg Pike, 4th Floor
Falls Church, VA 22041

Fax: **703/671-7937**; indicate "TAFIM" on cover sheet.

D.2 TAFIM CHANGE PROPOSAL SUBMISSION FORMAT

a. Point of Contact Identification

(1) Name:

(2) Organization and Office Symbol:

(3) Street:

(4) City:

(5) State:

(6) Zip Code:

(7) Area Code and Telephone #:

(8) Area Code and Fax #:

(9) E-mail Address:

b. Document Identification

(1) Volume Number :

(2) Document Title:

(3) Version Number:

(4) Version Date:

c. Proposed Change # 1

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

d. Proposed Change # 2

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

n. Proposed Change # n

(1) Section Number:

(2) Page Number:

(3) Title of Proposed Change:

(4) Wording of Proposed Change:

(5) Rationale for Proposed Change:

(6) Other Comments:

D.3 FORMAT GUIDANCE

The format in Section D.2 should be followed exactly as shown. For example, Page Number should not be entered on the same line as the Section Number. The format can accommodate, for a specific TAFIM document, multiple change proposals for which the same individual is the Point of Contact (POC). This POC would be the individual the TAFIM project staff could contact on any question regarding the proposed change. The information in the **Point of Contact Identification** part (**D.2 a**) of the format would identify that individual. The information in the **Document Identification** part of the format (**D.2 b**) is self-evident, except that volume number would not apply to the CMP or PMP. The proposed changes would be described in the **Proposed Change #** parts (**D.2 c, D.2 d, or D.2 n**) of the format.

In the **Proposed Change #** parts of the format, the Section number refers to the specific subsection of the document in which the change is to take place (e.g., Section 2.2.3.1). The page number (or numbers, if more than one page is involved) will further identify where in the document the proposed change is to be made. The Title of Proposed Change field is for the submitter to insert a brief title that gives a general indication of the nature of the proposed change. In the Wording of Proposed Change field the submitter will identify the specific words (or sentences) to be deleted and the exact words (or sentences) to be inserted. In this field providing identification of the referenced paragraph, as well as the affected sentence(s) in that paragraph, would be helpful. An example of input for this field would be: "Delete the last sentence of the second paragraph of the section and replace it with the following sentence: "The working baseline will only be available to the TAFIM project staff." The goal is for the commentor to provide proposed wording that is appropriate for insertion into a TAFIM document without editing (i.e., a line-out/line-in change). The D.2 c (5), D.2 d (5), or D.2 n (5) entry in this part of the format is a discussion of the rationale for the change. The rationale may include reference material. Statements such as "industry practice" would carry less weight than specific examples. In addition, to the extent possible, citations from professional publications should be provided. A statement of the impact of the proposed change may also be included with the rationale. Finally, any other information related to improvement of the specific TAFIM document may be provided in D.2 c (6), D.2 d (6), or D.2 n (6) (i.e., the Other Comments field). However, without some degree of specificity these comments may not result in change to the document.

This page intentionally left blank.